Minimally Interactive Segmentation with Application to Human Placenta in Fetal MR Images

Guotai Wang

A dissertation submitted in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

of

University College London.

Centre for Medical Image Computing

Department of Medical Physics and Biomedical Engineering

University College London

June 8, 2018

I, Guotai Wang, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the work.

Abstract

Placenta segmentation from fetal Magnetic Resonance (MR) images is important for fetal surgical planning. However, accurate segmentation results are difficult to achieve for automatic methods, due to sparse acquisition, inter-slice motion, and the widely varying position and shape of the placenta among pregnant women. Interactive methods have been widely used to get more accurate and robust results. A good interactive segmentation method should achieve high accuracy, minimize user interactions with low variability among users, and be computationally fast. Exploiting recent advances in machine learning, I explore a family of new interactive methods for placenta segmentation from fetal MR images. I investigate the combination of user interactions with learning from a single image or a large set of images. For learning from a single image, I propose novel Online Random Forests to efficiently leverage user interactions for the segmentation of 2D and 3D fetal MR images. I also investigate co-segmentation of multiple volumes of the same patient with 4D Graph Cuts. For learning from a large set of images, I first propose a deep learning-based framework that combines user interactions with Convolutional Neural Networks (CNN) based on geodesic distance transforms to achieve accurate segmentation and good interactivity. I then propose image-specific fine-tuning to make CNNs adaptive to different individual images and able to segment previously unseen objects. Experimental results show that the proposed algorithms outperform traditional interactive segmentation methods in terms of accuracy and interactivity. Therefore, they might be suitable for segmentation of the placenta in planning systems for fetal and maternal surgery, and for rapid characterization of the placenta by MR images. I also demonstrate that they can be applied to the segmentation of other organs from 2D and 3D images.

Impact Statement

The algorithms and software for image segmentation developed in this thesis have the potential to lead to benefits both inside and outside academia in a variety of ways.

First, they will benefit academic disciplines including machine learning and medical imaging. The dynamically balanced Online Random Forests developed in this thesis addresses a general problem of learning from imbalanced data with a changing imbalance ratio. It can be applied to not only interactive image segmentation, but also broader learning problems such as data stream classification in financial distress prediction. The proposed deep interactive segmentation methods address the problem of encoding user interactions with convolutional neural networks, which opens a new stream of research in interactive medical image computing with deep learning. These algorithms may also benefit researchers in the field of fetal growth assessment, modeling and surgical intervention. In the meanwhile, I contributed to the deep learning software for medical images NiftyNet[1], which may help researchers in the medical image computing community.

Secondly, researches in this thesis has the potential to benefit clinical practice and healthcare. The segmentation methods developed in this research may assist Radiologists to segment the placenta more efficiently. The new algorithms can potentially help them get accurate segmentation results with far less time and therefore reduce their burden. Surgeons may also benefit from the developed algorithms to obtain the segmentation results for better surgical planning, and clinicians may benefit from them for better diagnosis of the developing fetus. All these aspects have the potential to improve maternal and fetal healthcare. This impact could be brought about through

[1] http://niftynet.io

translational researches, collaborations with clinicians and commercial activities.

Thirdly, the research output may benefit industries such as medical imaging companies. This may help them develop better software for clinical use and novel products based on intelligent image computing. This impact could be brought about through patents and collaborations with technology transfer companies.

Acknowledgements

I would like to sincerely thank my primary supervisor Prof. Sébastien Ourselin, for his strong support, invaluable guidance and continuous encouragement throughout my PhD.

My sincere gratitude also goes to my secondary supervisor Tom Vercauteren and tertiary supervisor Maria A. Zuluaga and Juan Eugenio Iglesias for their very insightful advice and in-depth discussions during my research project.

I am also grateful to my clinical supervisor Prof. Jan Deprest for his suggestions and kind support. I would like to thank Anna L. David, Rosalind Pratt, Premal A. Patel, Michael Aertsen, Wenqi Li and Tom Doel for their kind help during my research.

I would like to thank as well my lab-mates in Translational Imaging Group, especially Luis, Marcel, Carla, Efthymios, Sebastiano, Michael, Yijing and Jieqing, who made our group an enjoyable working environment. I am also thankful to UCL for funding me with the Overseas Research Scholarship and Graduate Research Scholarship.

Last but not least, I am profoundly grateful to my family, my father Kaijia Wang, my mother Yuanxiu Shen, and my brother Jinguo Wang, for their extensive and endless support in all the past years. Special thanks from the bottom of my heart go to my wife Yuanyuan Nie, who shared all my pleasure and depression during these years.

List of Publications and Patent Applications

Peer-reviewed Journal Papers

1. **Guotai Wang**, Maria A. Zuluaga, Rosalind Pratt, Michael Aertsen, Tom Doel, Maria Klusmann, Anna L. David, Jan Deprest, Tom Vercauteren, and Sébastien Ourselin. Slic-Seg: A minimally interactive segmentation of the placenta from sparse and motion-corrupted fetal MRI in multiple views. *Medical Image Analysis*, 34:137-147, 2016.

2. **Guotai Wang**, Wenqi Li, Maria A. Zuluaga, Rosalind Pratt, Premal A. Patel, Michael Aertsen, Tom Doel, Maria Klusmann, Anna L. David, Jan Deprest, Sébastien Ourselin, and Tom Vercauteren. Interactive segmentation of medical images using deep learning with image-specific fine-tuning. *IEEE Transactions on Medical Imaging*, In press, 2018. DOI:10.1109/TMI.2018.2791721.

3. **Guotai Wang**, Maria A. Zuluaga, Wenqi Li, Rosalind Pratt, Premal A. Patel, Michael Aertsen, Tom Doel, Maria Klusmann, Anna L. David, Jan Deprest, Sébastien Ourselin, and Tom Vercauteren. DeepIGeoS: A deep interactive geodesic framework for medical image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, In press, 2018. DOI:10.1109/TPAMI.2018.2840695.

4. Eli Gibson, Wenqi Li, Carole Sudre, Lucas Fidon, Dzoshkun Shakir, **Guotai Wang**, Zach Eaton-Rosen, Robert Gray, Tom Doel, Yipeng Hu, Tom Whyntie,

Parashkev Nachev, Dean C Barratt, Sbastien Ourselin, M Jorge Cardoso, Tom Vercauteren. NiftyNet: A deep-learning platform for medical imaging. *Computer Methods and Programs in Biomedicine*, 158:113-122, 2018

Peer-reviewed Full-length Conference Papers

1. **Guotai Wang**, Maria A. Zuluaga, Rosalind Pratt, Michael Aertsen, Anna L. David, Jan Deprest, Tom Vercauteren, and Sébastien Ourselin. Slic-Seg: Slice-by-slice segmentation propagation of the placenta in fetal MRI using one-plane scribbles and online learning. In *MICCAI*, pages 29-37, 2015.

2. **Guotai Wang**, Maria A. Zuluaga, Rosalind Pratt, Michael Aertsen, Anna L. David, Jan Deprest, Tom Vercauteren, and Sébastien Ourselin. Minimally interactive placenta segmentation from motion corrupted MRI for fetal surgical planning. In *MICCAI Workshop on Interactive Medical Image Computing*, 2015

3. **Guotai Wang**, Maria A. Zuluaga, Rosalind Pratt, Michael Aertsen, Tom Doel, Maria Klusmann, Anna L. David, Jan Deprest, Tom Vercauteren, and Sébastien Ourselin. Dynamically balanced online random forests for interactive scribble-based segmentation. In *MICCAI*, pages 352-360, 2016.

4. Wenqi Li, **Guotai Wang**, Lucas Fidon, Sébastien Ourselin, M. Jorge Cardoso, and Tom Vercauteren. On the compactness, efficiency, and representation of 3D convolutional networks: brain parcellation as a pretext task. In *IPMI*, pages 348-360, 2017.

5. **Guotai Wang**, Wenqi Li, Sébastien Ourselin, and Tom Vercauteren. Automatic brain tumor segmentation using cascaded anisotropic convolutional neural networks. In *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries*, pages 178-190, Springer International Publishing, 2018.

Patent Applications

1. **Guotai Wang**, Sébastien Ourselin, Tom Vercauteren, Wenqi Li, Lucas Fidon. A system and computer-implemented method for segmenting an image. *U.K.*

Patent GB1709672.8, filed 16 June 2017.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The placenta plays a critical role in the growth and development of the fetus during pregnancy. Disorders of the placenta including abnormal placental structure and function are a cause of some diseases such as Twin-Twin Transfusion Syndrome (TTTS) [1, 2] and Intrauterine Growth Restriction (IUGR) [3]. They may also lead to poor maternal and fetal outcome including antepartum haemorrhage [4] and stillbirth [5]. With the advance of medical imaging techniques, such as Ultrasound and Magnetic Resonance Imaging (MRI), *in vivo* imaging of the placenta is becoming increasingly important, as it provides detailed structural and functional information for understanding the placenta. This supports a better assessment of fetal growth [6], more reliable diagnosis of fetal and maternal disease [7], and improved planning and guidance for fetal surgical treatment [8, 9]. Segmentation of the placenta from medical images allows quantitative measurements of the volume and shape of the placenta, which is desirable for placenta characterization, diagnosis, and surgical planning and guidance. Since automatic segmentation methods can rarely achieve sufficiently accurate and robust results for clinical use, taking advantage of user interactions to guide the segmentation attracts many attentions, and remains the state of the art for existing commercial surgical planning systems. However, most existing interactive methods do not work well on placenta images, or require a large amount of user interactions and increase burden on the user. This thesis will explore novel interactive methods to segment the placenta with high accuracy and a minimal amount of user interactions, and demonstrate their application to other 2D and 3D segmentation tasks.

This chapter starts with an introduction of placental anatomy and abnormality in 1.1, and reviews clinical imaging of the placenta in 1.2. The current trend of placenta segmentation is presented in 1.3. Objectives and challenges of this research are described in 1.4. Contributions of this thesis are listed in 1.5. In 1.6, I summarize the structure of this thesis.

1.1 Placental Anatomy and Abnormality

The placenta is an organ that attaches to the maternal uterine wall, and connects to the developing fetus by the umbilical cord. It starts to develop after the blastocyst is implanted into the maternal endometrium, and separates from the uterine wall during the last stage of labor. Vessels in the umbilical cord branch out over the surface of the placenta, and further divide to form an extensive arterio-capillary-venous system. The main functions of the placenta include providing oxygen and nutrients to the fetus, removing waste products from the fetus, immunity and endocrine function, etc. Therefore, the placenta plays a critical role in the growth and development of the fetus during pregnancy. Figure 1.1 shows an illustration of the placenta and the fetus.

The human placenta usually has a disc shape, with the center being the thickest, and edges being the thinnest. The average weight of the placenta at term is 508g, and there is a positive correlation between placental weight and fetal weight [11]. It has been documented that the placental volume in the second trimester can be used to predict birth weight [12], and a relation between placental weight and birth weight was found in previous studies [13]. The placental volume and weight can be an indicator of nutritional or environmental problems during pregnancy [13], and it has been proposed as part of a screening test for the prediction of growth-restricted babies [14].

A range of placental abnormalities lead to poor maternal and fetal outcome. They are also a major contributor of obstetric haemorrhage. Previous studies reported placental abnormalities accounted for more than one third of pregnancy-related deaths due to haemorrhage [4]. Abnormalities of the placenta include attachment disorders, abnormal placental volume, weight, blood flow and shape, among others [15].

Placental attachment disorders (a.k.a. morbidly adherent placentas) [18] are due

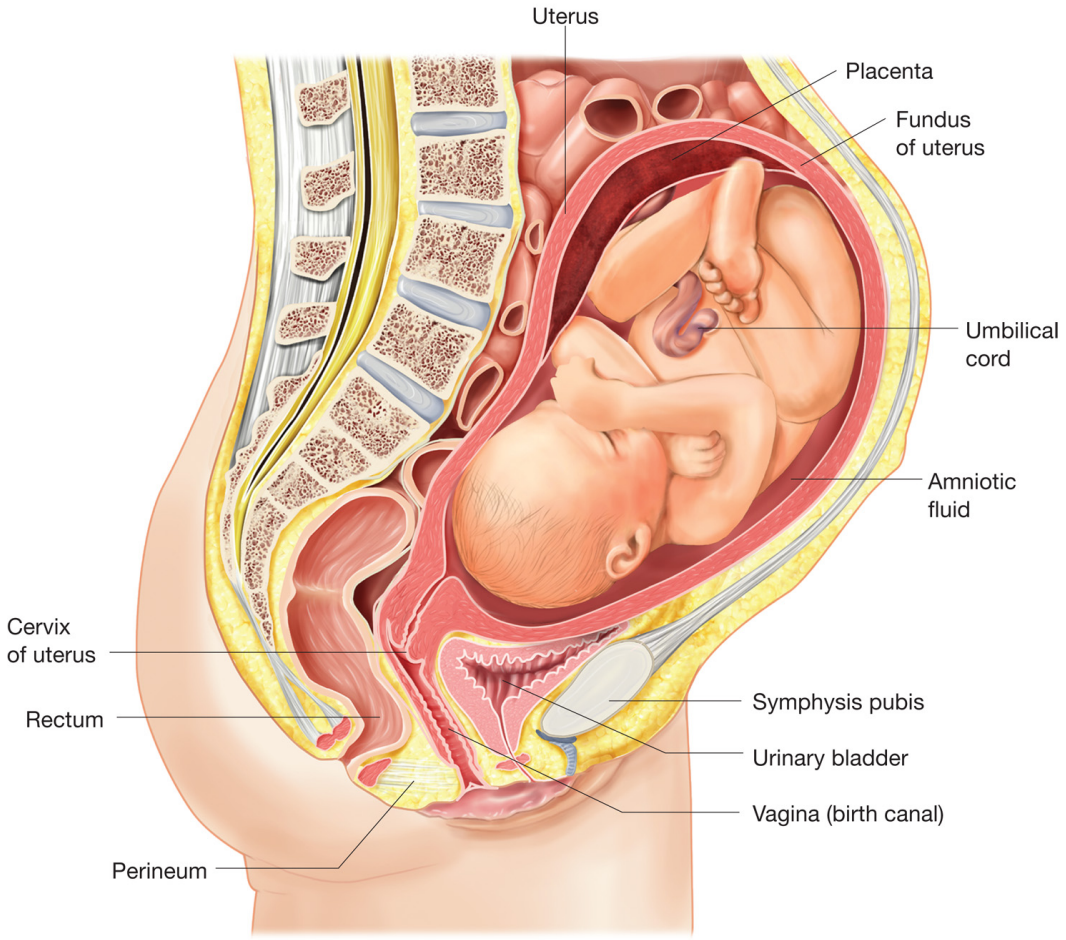


Figure 1.1: An illustration of the placenta and the fetus. Image from Biology Forums [online] [10].

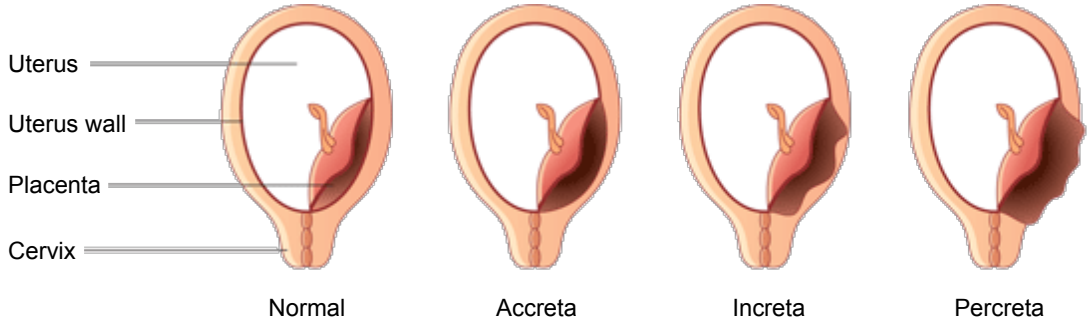


Figure 1.2: An illustration of normal placenta, placenta accreta, increta and percreta. Image from Singapore General Hospital [online] [16].

to an abnormally adherent placenta invading the myometrium, and are associated with life-threatening postpartum haemorrhage. The types of morbidly adherent placentas include placenta accreta, increta and percreta. Placenta accreta refers to the condition

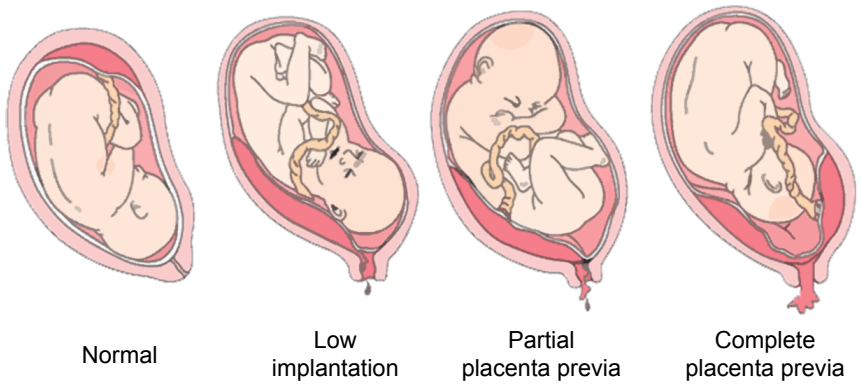


Figure 1.3: An illustration of normal placenta and different types of placenta previa. Image from Faculty of Medicine [online] [17].

in which the placenta is attached too deeply and too firmly into the uterus. In the case of placenta increta, the placenta is attached even more deeply into the muscle wall of the uterus. In the case of placenta percreta, the placenta grows through the uterus, sometimes extending to nearby organs such as the bladder. Figure 1.2 illustrates the normal placenta and placenta accreta, increta and percreta.

The position in the uterus where the placenta is attached has a large variation among different pregnant women. The placenta normally grows on the upper part of the uterus, and occasionally grows on the lower part of the uterus. Placenta previa is a condition where the placenta attaches to the lower part of the uterus and covers the cervix. In this case, there is a risk of bleeding during labor if the placenta is in front of the baby. Placenta previa can be categorized into three types: low implantation where the placenta implants in the lower portion instead of the upper portion of the uterus, partial placenta previa where a portion of the cervical orifice of the uterus is already covered by the placenta, and complete placenta previa where the placenta occludes the entire cervical orifice of the uterus. Figure 1.3 shows the normal placenta and different types of placenta previa.

Previous studies documented clinical associations with placental weight and fetal/placental weight ratio. For example, a large placenta may be related to maternal diabetes and a high placental weight is associated with a poor perinatal outcome [19]. A small placenta may be related to trisomies [19]. Common causes of unusually large placentas are villous edema, maternal diabetes mellitus, severe maternal anemia, fe-

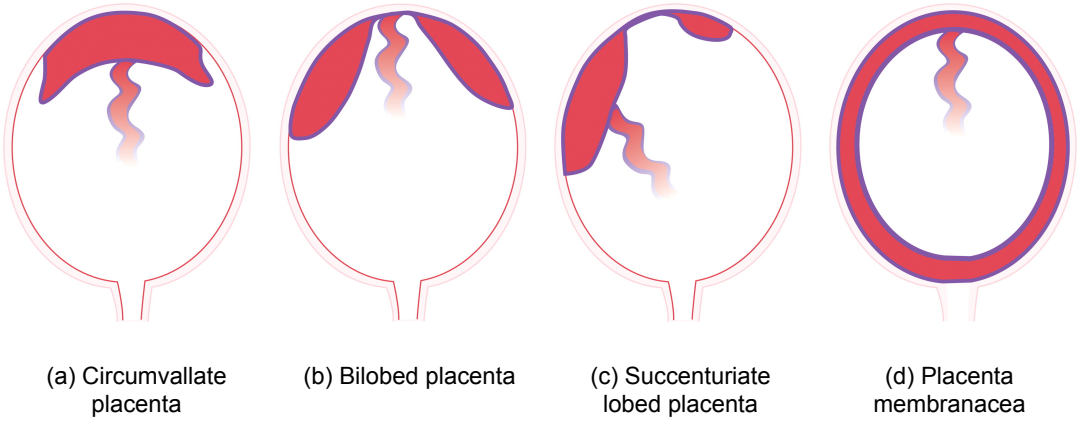


Figure 1.4: An illustration of placental shape abnormalities.

tal anemia, congenital syphilis, large intervillous thrombi, and a large blood clot beneath the chorionic roof of the placenta. Factors related to small placentas include low maternal weight before conception, low pregnancy weight gain, accelerated placental maturation and major fetal malformations. All these factors are associated with low maternal gestational blood volume expansion with resulting low blood flow from the uterus to the placenta. The most important risk factor is fetal growth retardation, i.e., IUGR, a condition where a fetus is unable to achieve its genetically determined potential size [20].

Some abnormalities of placental shape can lead to postpartum haemorrhage, e.g., circumvallate placenta, bilobed placenta, succenturiate lobed placenta, and placenta membranacea [15]. Circumvallate placenta is a condition where the fetal membranes create an edge of double folded membrane, as shown in Figure 1.4(a). There is an inward insertion of membranes from the edge towards the center of the placenta. This is often in association with a marginal infarction, haemorrhage, or fibrin deposition. Bilobed placenta occurs when the placenta is occasionally separated into two lobes, as shown in Figure 1.4(b). It increases the risk of vaginal bleeding during and after pregnancy. In the case of succenturiate lobed placenta, one or more small accessory lobes develop in the membranes at a distance from the main placenta, as shown in Figure 1.4(c). The accessory lobe may be retained in the uterus after delivery, causing serious haemorrhage. Placenta membranacea is a rare placental disorder characterized by the presence of fetal membranes (complete or partially) covered by chorionic villi,

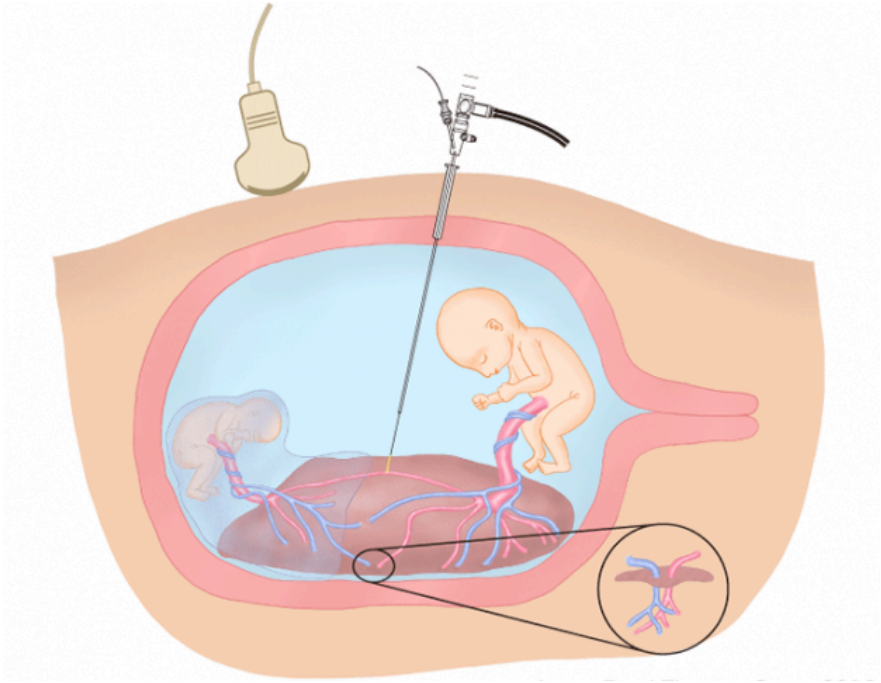


Figure 1.5: Laser photocoagulation of placental vessels in twin-twin transfusion syndrome. Image from Japan Fetal Therapy Group [online] [21].

which is illustrated in Figure 1.4(d). It may occasionally give rise to serious haemorrhage because of associated placenta previa or accreta.

In the case of twin pregnancy, some twin-specific anomalies of the placenta increase risks of birth defects, and can impact significantly perinatal morbidity and mortality [22]. In TTTS [1, 2], there is an important blood flow transfer through unidirectional arteriovenous anastomoses with an insufficient compensatory countertransfusion. Blood can be transferred from one twin (the donor) to the other (the recipient). This leads the donor to have decreased blood volume, restricting the growth of the donor. The recipient has increased blood volume, leading to higher risks of heart failure [23]. Intrauterine laser ablation has been an established surgical treatment for TTTS [8]. In this procedure, a fetoscope is used to image blood vessels on the surface of the placenta and the vessels found to connect the twins are coagulated using the laser. An illustration of laser photocoagulation of placental vessels in TTTS is shown in Figure 1.5.

Another twin-specific anomaly of the placenta is in the case of Selective Intrauterine Growth Restriction (SIUGR) [24]. Approximately 10% of monochorionic twins

encounter SIUGR because of uneven share of the placenta between the twins. In this case, one twin does not get enough oxygen and nutrients from the placenta. This leads to poor growth of that twin, i.e, selective growth restriction. SIUGR is increasingly considered to be an important complication of monochorionic twins and it has potential risks of Intrauterine Fetal Demise (IUFD) or neurological adverse outcome for both twins [25]. The term SIUGR is applicable in monochorionic pregnancies when the estimated fetal weight of the small fetus falls below the 10th percentile, which is widely accepted as a diagnostic criterion [26]. Current treatment methods of SIUGR in monochorionic twins include expectant management, cord coagulation or selective termination and laser photocoagulation [27].

1.2 Clinical Imaging of the Placenta

With the development of medical imaging, several modalities are now clinically available to image the developing fetus and the placenta, e.g., 2D and 3D Ultrasound (US), fetal MRI and others. Different modalities can provide complementary information with different contrast, resolution, field of view, etc.

1.2.1 Ultrasound

Ultrasound has been the primary imaging method for prenatal diagnosis of fetal anomalies. The creation of an ultrasound image involves three steps: producing sound waves, receiving echoes and reconstructing an image from these echoes. Typically, a piezoelectric transducer produces a sound wave the frequency of which can range from 1 to 18 MHz. Superficial structures are usually imaged at a higher frequency with better axial and lateral resolution while deeper structures are imaged at a lower frequency with greater penetration [29]. After sound waves are transmitted into the body, they are partially reflected at the interlayer between tissues with different acoustic impedance or scattered from small structures. Some reflected waves return to the transducer and lead to vibrations of the transducer. The transducer turns the vibrations into electric pulses and transforms them into a digital image. With the received echoes, the scanner can determine the strength of each echo and the time it took the echo to be received from when it was transmitted. Therefore, the scanner can locate the pixel in the im-

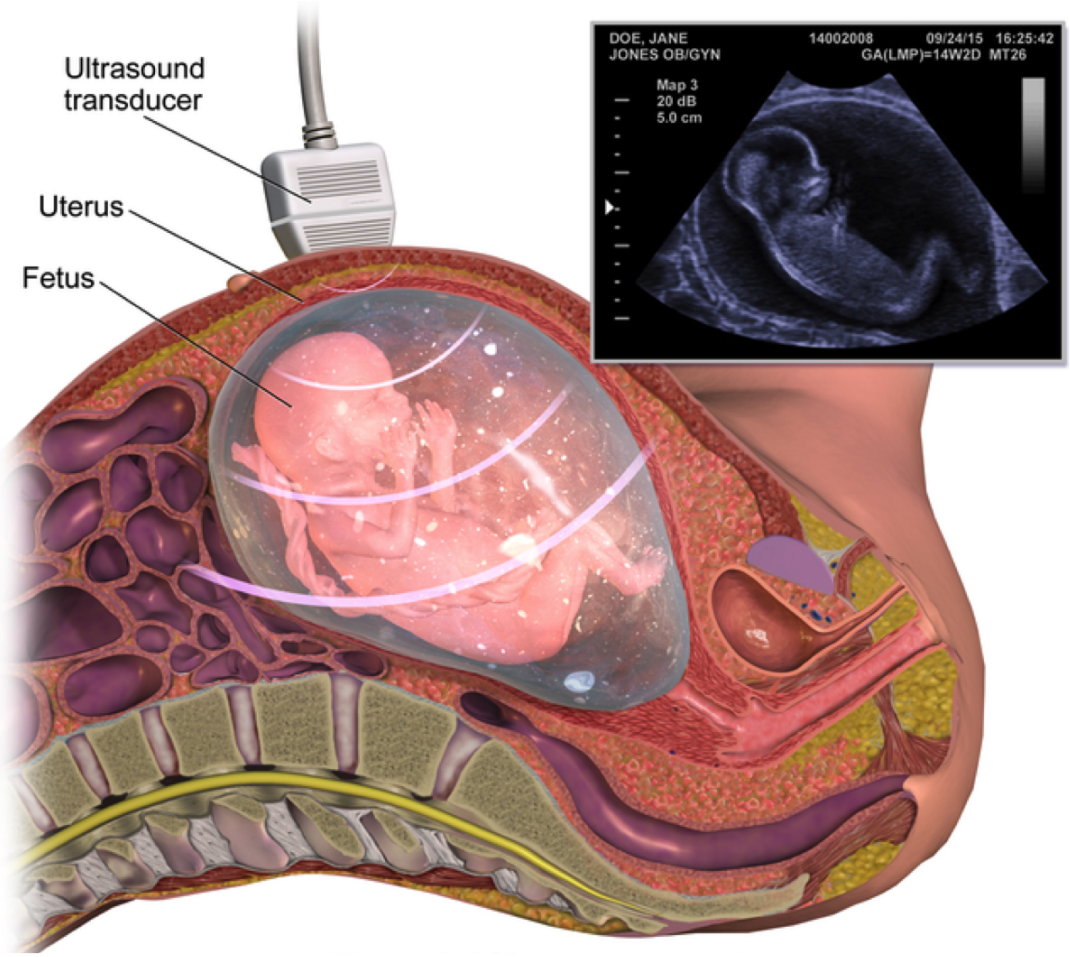


Figure 1.6: An illustration of fetal ultrasound. Image from Wikipedia [online] [28].

age related to a reflected echo and determine its intensity [30]. An illustration of fetal ultrasound imaging is shown in Fig. 1.6.

Several modes of ultrasound can be used in medical imaging. A-mode allows scanning a line through the body. B-mode or 2D mode scans a plane through the body, which is most commonly used. Another important mode of ultrasound imaging is the Doppler ultrasound. It makes use of the Doppler shift principle and can reflect the direction and velocity of blood flow. It is most widely used in the detection of fetal cardiac pulsation and pulsations in various fetal blood vessels including those in the placenta [31]. In addition, 3D ultrasound can provide a three-dimensional imaging of the fetus, and it allows the operator to obtain views that might not be available using ordinary 2D scanning. 3D ultrasound can provide better volumetric measurements of fetal organs, such as the fetal lungs [32], the fetal heart [33] and the placenta [34]. Other

ultrasound imaging techniques are also used for different contrasts, such as Contrast-Enhanced Ultrasound (CEUS), ultrasound molecular imaging, elastography [35] and compression ultrasonography [36], etc.

For fetal growth assessment, fetal ultrasound can be used for gestational age determination and fetal size measurement. For example, the measurement of crown-rump length can be made in early pregnancy to obtain an accurate estimation of the gestational age [37]. The measurement of biparietal diameter and femur length can also be used for dating at early stage of pregnancy [38]. The abdominal circumference is an important measurement to make in late pregnancy and it reflects fetal size and weight, which is useful in fetal growth monitoring [6]. Many structural abnormalities in the fetus can also be reliably diagnosed by an ultrasound scan. Common examples include spinal bifida, hydrocephalus, duodenal atresia and congenital cardiac abnormalities [39]. Fetal ultrasound has become an effective tool for localization of the site of the placenta and determining its lower edges. It can be used to make a diagnosis or an exclusion of placenta previa [40].

Ultrasound imaging has a lot of advantages. It is widely available and inexpensive compared with MRI. It can provide realtime imaging with high spatial resolution with high frequency transducers. It is safe without ionizing radiation and rarely causes discomfort to the patient. However, ultrasound imaging has low soft-tissue contrast and small field of view. It performs very poorly when there is an extreme difference in acoustic impedance. The image quality is corrupted by noises and artifacts. In addition, ultrasound imaging is operator-dependent, and the acquisition of good-quality images needs a high level of skill and experience.

1.2.2 Fetal MRI

With advantages such as large field of view, lack of ionizing radiation and good soft tissue contrast, MRI is widely used for diagnosis and surgical planning for adults. In the past two decades, fetal MRI has emerged as a clinically useful supplement to ultrasound and is increasingly used for prenatal and perinatal management [41]. Fetal MRI has advantages in demonstrating pathology of the fetal brain, fetal lungs, complex syndromes, and conditions associated with reduction of amniotic fluid [42].

Similar to MR imaging of adults, different protocols can be used for imaging of the fetus [42]. However, the complex pattern of motion during fetal scanning leads to several specific protocols for fetal MRI [43]. Basically there are two types of motion: maternal motion and fetal motion. With regard to the former, fetal MRI faces similar problems as abdominal MR imaging of adults. Fetal movements include fetal bulk motion, fetal extremity movements and internal fetal movements (e.g, heart beat). The fetal bulk motion interferes a lot with image quality.

Single-shot Fast Spin-echo (SSFSE) T2-weighted imaging is standard in fetal MRI. This sequence provides a stack of 2D slices by using a single excitation pulse that is followed by a rapid train of refocused echoes, providing all the data needed for a 2D slice. Because the center of k-space is sampled within a fraction of a second, the intra-plane motion is essentially frozen and motion-induced artifacts in 2D slices are nearly absent. However, the motion can still occur between neighboring slices and corrupt the 3D volume. There is also a widespread use of Half-Fourier Single-shot Turbo Spin-echo (HASTE) T2-weighted sequences, which is similar to SSFSE. They can excellently depict the fetal brain, fluid filled cavities, the fetal lungs and the placenta, etc [42, 44].

Several kinds of T1-weighted sequences have also been used in fetal MRI. Fast Low Angle Shot (FLASH) sequences are the most robust ones. Though T1-weighted sequences provide little information over the T2-weighted SSFSE sequences, they are suitable for detection of haemorrhage, calcification, fat deposition and fetal organs with high T1-hyperintensity (e.g, the thyroid and the liver) [45].

Fetal MRI can also be performed using balanced Steady-state Free Precession (SSFP) sequences as balanced Fast Field Echo (b-FFE) [46]. Balanced SSFP sequences differ from standard gradient echo sequences by reusing transverse magnetization to form a steady-stage magnetization. They are extremely fast,6 and provide good signal with the contrast being a mixture of T1 and T2. B-FFE is a preferred sequence for visualization of the fetal heart and vessels [47].

Other MRI techniques such as Diffusion Weighted Imaging (DWI) and Magnetic Resonance Spectroscopy (MRS) have also been applied to fetal MRI. DWI has been

used for anatomic characterization of fetal brain development [48] and study of placental insufficiency related to IUGR [49]. MRS has been used for assessment of normal fetal brain maturation [50] and placental metabolism [51]. However, these protocols require long scan time and have low image resolution.

1.2.3 Other Modalities

In addition to ultrasound and MRI that are not ionizing, low dose fetal Computed Tomography (CT) can be used in the prenatal evaluation of skeletal abnormalities [52]. However, it carries a risk of fetal exposure to radiation. Some invasive techniques can also be employed for imaging of the fetus and the placenta, such as fetoscopy and photoacoustic imaging. However, photoacoustic imaging has never been demonstrated in the clinic.

Fetoscopy is an endoscopic procedure that uses one specific type of endoscope, i.e., fecoscope, to look into the uterus and allows access to the fetus, the amniotic cavity, the umbilical cord and the placenta. Fetoscope is often used with ultrasound for image guided interventional procedures, such as the intrauterine treatment of tracheal occlusion [53], fetal blood sampling and imaging of the placenta for laser ablation in TTTS [54]. However, fetoscope has limited ability to image the vasculature beneath the placental surface due to strong light scattering in biological tissues [55].

In order to achieve better visualization of the placental vasculature, researchers have started to investigate the use of photoacoustic techniques for imaging of the placenta [55]. In photoacoustic imaging [56], laser pulses are delivered into biological tissues. The tissues absorb part of the delivered energy and expand as a result of heating. The transient thermoelastic expansion leads to ultrasonic emission, which is detected by ultrasonic transducers to produce images. Since blood usually has orders of magnitude higher absorption than surrounding tissues, photoacoustic imaging can provide good visualization of blood vessels. It has the potential to monitor placenta oxygenation [57] and assist minimally invasive fetal surgeries [55].

1.3 Segmentation of the Placenta

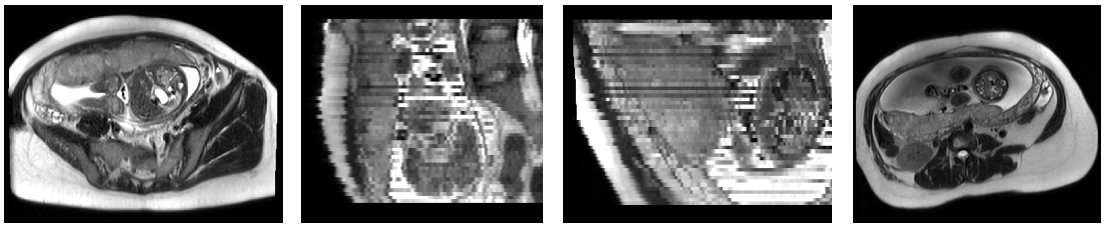
The task of placenta segmentation is to extract the placenta from medical images such as ultrasound and MR images. An accurate segmentation result can provide reliable measurements of the volume and 3D shape of the placenta. This can help the assessment of fetal growth (e.g, estimating fetal weight [19]) and diagnosis of placental abnormality (e.g, bilobed placenta and succenturiate lobed placenta [58]). A segmentation can also be used to model the variability of shapes of the placenta [59] or predict postnatal outcome [60]. In image-guided intrauterine fetal surgeries such as laser ablation therapy of TTTS, a fusion of endoscopic image mosaics with a 3D model of the placenta helps to improve planning and guidance in the surgical treatment [61, 62, 9]. The 3D model requires an accurate segmentation of the placenta from 3D Ultrasound or fetal MR images.

Previous works of placenta segmentation focused on dealing with ultrasound images. For example, in [63], a random walker method was used to segment the placenta from 3D ultrasound. The marching cubes algorithm was used in [62] to segment 3D ultrasound for image guidance of fetal surgery. A deep learning method was used in [64] to segment the placenta of the first trimester from 3D ultrasound.

Despite previous works for ultrasound-based placenta segmentation, fetal ultrasound has a limited field of view. It is hard to image the entire placenta in a single ultrasound frame or volume, especially for the second and third trimester. In addition, speckle noise and low contrast of fetal ultrasound have a negative impact on the segmentation accuracy. In contrast, fetal MRI can provide better soft tissue contrast and larger field of view, which makes it possible to capture the entire placenta and fetus even in a large gestational age. However, segmentation of the placenta from fetal MR images has rarely been studied.

1.4 Objectives and Challenges

This thesis aims to segment the placenta from fetal MR images for fetal surgical planning or characterization of the placenta. However, accurate placenta segmentation from fetal MR images is a challenging task due to several reasons.



(a) Axial view (b) Sagittal view (c) Coronal view (d) Axial view

Figure 1.7: Examples of fetal MR images. (a), (b) and (c) are from one patient while (d) is from another. Note the motion artifacts and different appearances in odd and even slices in (b) and (c). The position of the placenta is anterior in (a), but posterior in (d).

First, some inherent challenges in medical images make accurate segmentation difficult to achieve. They include imaging noise, ambiguous boundaries as a result of partial volume effect and low contrast. In addition, bias field inconsistency commonly exists in MRI and leads to nonuniform intensity and spatial changes in tissue statistics, i.e., mean and variance [65].

Secondly, differently from regular adult MRI, fetal MRI suffers from low 3D image quality due to large inter-slice spacing and movement of the fetus. In order to reduce the scan time and avoid slice cross-talk artifacts, contiguous slices are not acquired sequentially, but in an interleaved manner with large inter-slice spacing, typically 3-4mm. Free movement of the fetus in the uterus during the scanning can cause severe motion artifacts [66]. Imaging protocols such as SSFSE allow the motion artifacts to be nearly absent in each slice, but inter-slice motion still corrupts the volumetric data. The interleaved acquisition leads to different appearances between neighboring slices. Fig. 1.7 shows some examples of fetal MR images, where (a), (b) and (c) are axial, sagittal and coronal views of the same acquisition, respectively. The data has a high 2D resolution in axial view. However, the image quality in sagittal and coronal view is very poor. In Fig. 1.7(b) and (c), there is a low resolution. It can be observed that the appearance is inhomogeneous among different slices, and the motion between neighboring slices additionally corrupts the image quality. Although some novel reconstruction techniques [67, 65, 68, 69, 70] can obtain super-resolution volumetric data with better image quality from sparsely acquired slices, they were mainly developed for the fetal brain. These methods have yet to demonstrate their utility for

placental imaging and require a dedicated non-standard acquisition protocol.

Thirdly, the placenta has a considerable variation of position and shape among different patients. For example, Fig. 1.7(a) shows the placenta is anterior in one patient but in Fig. 1.7(d) the placenta is posterior in another patient. The shape variation is also complex as shown in Fig. 1.4. Such complex variations make it hard to use statistical prior-knowledge such as shape/appearance models or propagated atlases [71, 72].

These issues make automatic segmentation of the placenta from fetal MR images very difficult. Previous works on medical image segmentation have shown that leveraging user inputs helps to obtain more precise segmentation results [73]. Motivated by these observations, this thesis investigates developing interactive methods to address the segmentation challenges where interventions given by the user can improve the accuracy of the placenta segmentation. However, requiring a large amount of user interactions can increase burden on the user. A good interactive segmentation method should require as few user interactions as possible, leading to interaction efficiency.

Thus, the objective of this thesis is to develop minimally interactive methods for segmentation of the placenta from fetal MR images so that accurate segmentation results can be obtained with a minimal amount of user interactions.

1.5 Thesis Contribution

This thesis focuses on developing interactive methods for placenta segmentation from fetal MR images. I investigate leveraging machining learning techniques to obtain high accuracy with a minimal amount of user interactions. Contributions of this thesis are summarized as following:

- A family of novel interactive segmentation methods based on state-of-the-art machine learning techniques (Random Forests and Deep Learning) are proposed for placenta segmentation from 2D slices, 3D volumes and multiple volumes of the same patient (4D), respectively.

- An Online Random Forests (ORF)-based interactive segmentation method is proposed to segment the placenta from 2D or 3D fetal MR images. To address the problem with imbalanced and gradually given scribbles, ORFs are extended

to generic Dynamically Balanced Online Random Forests (DyBa ORFs) that are more suitable than existing ORFs for scribble-based interactive segmentation.

- A minimally interactive framework (Slic-Seg) dealing with a single and multiple motion-corrupted fetal MR volumetric images is proposed. It only requires user interactions in a single slice to segment one volume and is able to refine an initial segmentation using inter-slice and inter-image consistency based on co-segmentation with 4D Graph Cuts.

- Two deep learning-based frameworks for interactive segmentation are developed. The first framework (DeepIGeoS) combines Convolutional Neural Networks (CNNs) and user interactions that are used for inference, and it improves the efficiency and reduces the user interaction time for accurate segmentation. The second framework (BIFSeg) proposes image-specific fine-tuning to improve segmentation accuracy and can segment previously unseen objects, which reduces the requirement of annotations for training and is adaptive to a specific test image.

1.6 Thesis Structure

This chapter gives an introduction of placenta segmentation, which summarizes the clinical background, objectives, challenges and contributions of this research.

Chapter 2 gives a literature review of state-of-the-art works for segmentation of medical images, including fetal MR images.

Chapter 3 deals with segmentation of the placenta from a 2D slice based on learning from user-provided scribbles. To deal with imbalanced training data with a changing imbalance ratio, I propose a dynamically balanced ORF and apply it to interactive segmentation [74].

In chapter 4, I propose a framework to segment the placenta from a volumetric fetal MR image with a minimal amount of user-interactions using ORFs and slice-by-slice propagation. This is the first work to apply online learning to segment motion-corrupted fetal MR images [75]. In addition, I propose a probability-based 4D Graph

Cuts method to deal with multiple motion-corrupted volumes of the same patient, and this leads to improved segmentation accuracy [76].

In chapter 5, I investigate the application of CNNs to interactive segmentation and propose a new deep learning-based framework named as DeepIGeoS [77]. The framework includes combining user-interactions with CNNs through geodesic distance transforms, resolution-preserving networks, and a Conditional Random Field (CRF) that can be jointly trained with CNNs.

In chapter 6, another deep learning-based interactive segmentation method called BIFSeg is proposed. It combines CNNs with bounding boxes and optional scribbles with image-specific fine-tuning. The proposed method allows a trained model to deal with previously unseen objects.

Chapter 7 concludes this thesis, and discusses future works and applications beyond fetal MR images.

Chapter 2

Literature Review

This section reviews state-of-the-art works on segmentation of medical images, including fetal MR images. Section 2.1 and Section 2.2 give an introduction of automatic and interactive segmentation methods, respectively. Section 2.3 reviews conditional random fields and graph cuts used in image segmentation, and Section 2.4 reviews co-segmentation methods. Section 2.5 introduces some basics of deep convolutional neural networks. In Section 2.6, related works on fetal MR image segmentation are reviewed.

2.1 Automatic Segmentation of Medical Images

This section classifies automatic image segmentation methods into four main categories: segmentation with low-level features, segmentation with active contours, segmentation with prior models and segmentation with machine learning. More detailed overviews of automatic image segmentation techniques can be found in [78, 79].

2.1.1 Segmentation with Low-level Features

Using low-level features such as pixel intensity and edges is one of the simplest and fastest segmentation methods. A typical algorithm in this category is thresholding [80], which assumes the image is formed from regions with different gray levels. It uses a threshold function to segment the image into the foreground and the background. For

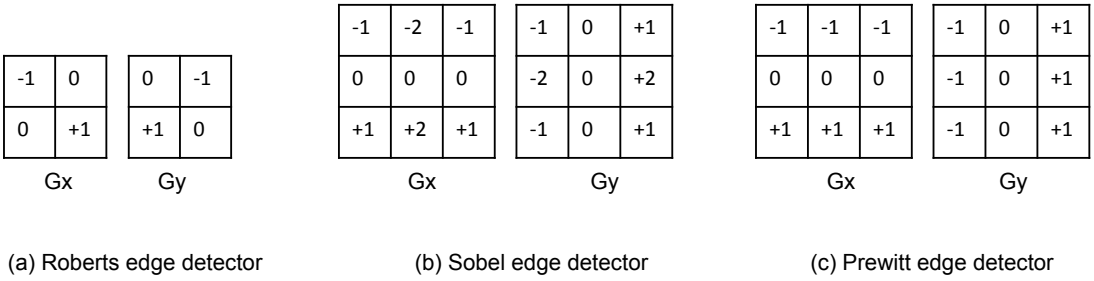


Figure 2.1: Examples of kernels of edge detectors.

example:

$$g(x,y) = \begin{cases} \text{foreground}, & \text{if } X(x,y) \geq T \\ \text{background}, & \text{if } X(x,y) < T \end{cases} \tag{2.1}$$

where $X(x,y)$ is the pixel intensity at position (x,y) and T is a threshold value. Finding a proper threshold value is difficult for many medical images. The Otsu's thresholding proposes to find the threshold value automatically based on minimizing the intra-class variance [81]. A global single thresholding can hardly provide satisfactory results for images with inhomogeneous appearance. Adaptive local thresholding and multi-level thresholding are proposed to deal with more complex image contexts [82].

Edge detection methods rely on local changes of intensity along object boundaries for segmentation. To extract the edge, typically an edge detector is used, such as Roberts detector, Sobel detector, Prewitt detector and LoG edge detector [83]. Three examples of edge detectors are shown in Fig. 2.1. Due to noise and low contrast of medical images, these simple detectors using local gradient can hardly achieve accurate results. More advanced edge detection methods such as mathematical morphology [84] and neural networks [85] considering a larger image context are proposed to reduce false positives and better enhance edges.

Region growing [86] starts the segmentation with a seed point and checks adjacent pixels against a predefined homogeneity criterion. Pixels meeting the criterion are added to the region. Repetitive applications of the criterion lead to a growth of the segmented region. Intensity threshold and gradient magnitude are often used as the growing criterion [87]. However, the segmentation result depends on the selected seed

point in many applications. Instead of being selected manually, the seed point can be automatically located based on image texture [88]. In addition, the homogeneity criterion can be made adaptive to local image context to improve the segmentation performance on medical images [89].

2.1.2 Segmentation with Active Contours

Active contours [90] are curves defined within an image domain that deform under the influence of internal and external forces. The internal force depends on the curve itself while the external force is based on the image context. These forces are defined so that the curve can conform to an object boundary, which gives a segmentation result. A typical example of active contours is the snakes model [91], which is a curve $\mathbf{v}(s) = [x(s),\ y(s)]$, $s \in [0,1]$. The segmentation process is an energy minimization problem defined as following:

$$E = \int_0^1 E_{int}(\mathbf{v}(s)) + E_{ext}(\mathbf{v}(s))ds \tag{2.2}$$

where E_{int} is the internal energy based on the first and second derivatives of $\mathbf{v}(s)$ with respective to s:

$$E_{int}(\mathbf{v}(s)) = \frac{1}{2}\Big(\alpha|\mathbf{v}'(s)|^2 + \beta|\mathbf{v}''(s)|^2\Big) \tag{2.3}$$

where α and β are weighting parameters for the curve's tension and rigidity, respectively.

$$E_{ext}(\mathbf{v}(s)) = -|\nabla X(\mathbf{v}(s))|^2 \tag{2.4}$$

where ∇ is the gradient operator. Snakes have two main limitations: they are sensitive to initialization and cannot cope with boundary concavities and topological changes. The Gradient Vector Flow (GVF) was proposed to obtain a better performance to deal with boundary concavities [90].

The level set method was proposed to deal with complex topological changes for active contours [92]. It represents the curve or surface as the zero iso-contour of a

function ϕ that evolves at different time:

$$\phi(\mathbf{v}(t),t) = 0 \tag{2.5}$$

Taking the time derivative on both sides of the equation leads to:

$$\phi_t + \nabla\phi(\mathbf{v}(t),t) \cdot \mathbf{v}'(t) = 0 \tag{2.6}$$

This yields an evolution equation for ϕ:

$$\phi_t + F|\nabla\phi| = 0 \tag{2.7}$$

Where F is the speed function $F = \nabla\phi \cdot \mathbf{v}'(t)/|\nabla\phi|$. It can be redefined by different variants [93, 94, 95].

2.1.3 Segmentation with Prior Models

Methods that incorporate high-level knowledge such as a-priori information have proven to be more stable against local image artifacts and perturbations than conventional low-level algorithms [72].

Shape models are often used to constrain active contours. The Active Shape Model (ASM) [96] learns patterns of shape variability from a training set, and allows the prior shape to deform to fit a test image in ways consistent with the training set. The shape model can be represented implicitly by a signed distance function to drive level set evolution [97]. Several techniques such as Gaussian distribution modeling, manifold learning or sparse representation have been used to model shape variations [98]. Active Appearance Models (AAMs) are extended from ASMs where shape and intensity of an image patch are integrated into a statistical model [99].

Atlas-based segmentation methods use one or multiple pre-labeled images (atlases) to propagate the labels to new images by registration [100]. Multi-atlas label fusion methods have been extensively studied in recent years [101]. Such methods register each atlas with labels to a target image and obtain the label of the target image by fusing labels of the atlases. Label fusion can be done with several strategies, such as

majority voting, atlas selection, simultaneous truth and performance level estimation, locally weighted label fusion and joint label fusion [102, 103].

2.1.4 Segmentation with Machine Learning

Machine learning methods learn patterns from a set of data and use them to guide the segmentation. These methods include unsupervised learning and supervised learning. Unsupervised methods do not need annotated training images, and work like density estimation in statistics or clustering to summarize and present data by their main features. Supervised methods require a set of training images with their corresponding labels.

2.1.4.1 Segmentation with Unsupervised Learning

The K-means algorithm is one of the most popular unsupervised learning methods [86]. It partitions the image into K clusters based on the mean of each cluster, i.e., each pixel is assigned to the cluster with the nearest mean based on Euclidean distance. The user should select the value of K to segment the image. The segmentation might be sensitive to outliers, initial values and noise. Some derivative methods have been proposed to address these problems [104]. Fuzzy C-mean is an extension of K-means. It uses a fuzzy partition rather than a hard partition, i.e., a pixel is partitioned to a cluster with a probability. It has proven useful in producing good segmentation for images with noise and intensity inhomogeneity [105].

Mixture models solved with Expectation Maximization (EM) is also a widely used unsupervised learning method [86]. It is an iterative process to calculate a maximum-likelihood estimation. In the first step (E step), the expectation of likelihood is calculated. In the second step (M step), the maximum-likelihood estimation is calculated. The iteration continues until the stop condition is true. The EM algorithm is often used to estimate a Gaussian Mixture Model (GMM) of the observed image intensity [106]. It can also be combined with atlases with a spatial prior for higher segmentation accuracy [107].

The Auto-Encoder (AE) is one type of unsupervised learning with deep neural networks [108]. A basic AE has two parts including an encoder and a decoder. The

encoder maps an input to a hidden representation, and the decoder maps the hidden representation to a reconstructed version of the input. AE can learn high-level features automatically from a dataset without supervision, and such features can be used for segmentation tasks. In [109], stacked AEs were used to automatically learn deep feature representations of Cryosection brain images, and these features were sent to a Softmax classifier for segmentation. In [110], deep feature representations learned by stacked AEs were used to guide atlas-based new infant brain segmentation from MR images.

2.1.4.2 Segmentation with Supervised Learning

Supervised learning for segmentation uses a classifier that is learned from a set of training images with annotations. After learning, the classifier is used to segment new images. Many algorithms can be used for classification, and some typical examples are k-Nearest Neighbor, Bayesian classifier, decision trees, neural networks, Support Vector Machine (SVM), etc [111].

Random Forest (RF) [112] is one of the most widely used learning methods for medical image segmentation, and has proven to be efficient with competitive performance. A random forest is a set of decision trees. Traditional decision trees have shown problems related to over-fitting and lack of generalization. Random forests mitigate such problems by introducing randomness in the training stage and combining the output of multiple randomized trees in the testing stage [113]. In [114], entangled decision forests were proposed to capture long-range semantic context for segmentation, where the binary tests at each tree node depend on the results of tests applied earlier in the same tree and at image points offset from the voxel to be classified. In the GeoF method [115], generalized geodesic distance transforms of probability maps were used as extra features (a.k.a auto-context) for RFs to achieve spatially consistent semantic image segmentation.

Recently, deep learning techniques with CNNs are achieving increasing success in image segmentation [116]. CNNs can find the most suitable features through automatic learning instead of manual design. Typical CNNs such as AlexNet [117], GoogleNet [118], VGG [119] and ResNet [120] were originally designed for image

classification tasks. Some early works adapted these networks for pixel labeling with patch or region-based methods [121, 122]. Such methods achieved higher accuracy than traditional methods that relied on hand-crafted features, but they suffered from inefficiency for testing.

In [123], a Fully Convolutional Network (FCN) was proposed to take an entire 2D image as input and obtain a dense segmentation. In order to overcome the problem of potential loss of spatial resolution due to multi-stage max-pooling and downsampling, it uses a stack of deconvolution (a.k.a. upsampling) layers and activation functions to upsample the feature maps. Inspired by the convolution and deconvolution framework of FCNs, a U-shape network (U-Net) was proposed for 2D biomedical image segmentation [124].

DeepLab [125] is another state-of-the-art 2D CNN for semantic segmentation. It uses dilated convolution (a.k.a. atrous convolution) [126] to enlarge the receptive field of convolution kernels, so that the network can capture larger context without increasing the number of parameters. DeepLab also proposes convolution at multiple sampling rates so that image context at multiple scales can be integrated for better performance. In addition, it uses a fully connected Conditional Random Field (CRF) [127] for spatial regularization.

For 3D segmentation, DeepMedic [128] was proposed to segment image patches with a dual pathway that processes the input image at multiple scales. A 3D fully connected CRF was proposed for post-processing. However, the patch-based segmentation limits the size of context that can be used, and has low efficiency during testing. In [129], the U-Net was extended to its 3D version and used to learn from a sparsely annotated training set. In [130], a similar structure called V-Net was proposed to segment the prostate from 3D MR images.

HighRes3DNet [131], of which I am a co-author, is a high-resolution, compact convolutional network for volumetric image segmentation. The network uses dilated convolution to preserve resolution of 3D feature maps, and employs residual connections to improve the training speed. It has achieved state-of-the-art performance for brain parcellation with fewer parameters than 3D U-Net [129] and V-Net [130].

2.2 Interactive Segmentation Methods

Interactive segmentation methods have been widely used [73]. They provide a balance between manual delineation, which often gives accurate and robust results with long segmentation time, and automatic segmentation, which saves time for user interactions but often lacks robustness. In practical applications, an interactive segmentation method should achieve high accuracy, minimize user interactions, be computationally fast and achieve low variance of results obtained by different users. For interactive segmentation, user interactions can be given in several ways including seed points [132], scribbles [133, 134, 135], bounding boxes [136] and others [137, 73].

2.2.1 Interactive Segmentation without Machine Learning

A series of traditional interactive segmentation methods define some specific rules to generate the segmentation based on user interactions. The seeded region growing [132] expands the segmented region of an object from seeds based on the gray value of pixels. User-guided 3D active contour segmentation [138, 90] employs the user inputs as seeds or initial contours of the target organ, and defines the external forces of active contours based on image gradient. Live-wire [139] uses gradient information to compute optimal boundaries as the user moves the mouse starting form a manually specified seed point. Geodesic Framework [134] and GeoS [140] classify a pixel based on its weighted geodesic distance to scribbles. The Random Walks method [135] assigns a pixel with the label for which a random walker is most likely to reach first. GrowCut [141] uses scribbles to set the initial state of a cellular automation for the pixel labeling task of a 3D image. These methods are popular as a general framework for many applications, but require a large amount of user interactions (e.g., user-provided scribbles or seed points) to get accurate results.

2.2.2 Interactive Segmentation using Machine Learning

Machine learning methods have been widely used to improve the segmentation performance and reduce the amount of user interactions. Graph Cuts [133] takes user-provided scribbles as hard constraints, and uses them to estimate intensity distributions of the foreground and the background, which is often based on GMMs [133, 142, 143].

GrabCut [136] uses iterated Graph Cuts for foreground extraction. It requires the user to provide a bounding box for the target, and iteratively updates a GMM and the Graph Cuts, leading to reduced user interactions compared with the original Graph Cuts method. The 4D Active Cut [144] actively selects candidate regions based on the segmentation confidence for querying the user, without the need to refine the segmentation slice by slice. In [145], intelligent scribble guidance based on logistic regression was proposed to reduce user interactions. TurtleSeg [146] applies active learning to 3D medical image segmentation by constructing an "uncertainty field" in order to alleviate the user from choosing where to provide interactive input.

Despite their success in many applications, most of the above-mentioned interactive methods rely on low dimensional features, and need many user interactions to deal with images with low contrast and weak boundaries. To tackle this problem, algorithms based on high-level features have been proposed to get more accurate segmentation with fewer user interactions. High-level features are often combined with machine learning methods for better distinguishing different types of tissues in medical images. For example, in [147], an SVM classifier using intensity and Gabor features was trained on user-selected seed points for tumor and ventricle segmentation from brain MR images.

Random Forest can be efficiently used for interactive segmentation, as it is very fast to compute while yielding state-of-the-art performance in machine learning and vision problems. In [148], RFs were used for interactive texture segmentation, where the RFs learn from user-provided scribbles in an image and then predict the label of the remaining pixels. The ImageJ/Fuji software provides a plugin called Trainable Weka Segmentation [149] that uses classifiers including RFs to learn from user inputs and classify the remaining pixels. Ilastik [150] also uses RFs to learn from labels provided by the user with a set of nonlinear features. It can provide realtime feedback to allow interactive refinement of the result. The Super-Region Volume Segmentation method (SuRVoS) [151] partitions a volumetric image into hierarchical segmentation layers (named super regions), and learns from user inputs to label the rest of the volume with RFs, SVM, Gradient Boosting and other algorithms. In [152], RFs were combined with

various image features extracted from multiple scales to segment the coronary artery from 3D computed tomography angiography images. That method was shown to require limited user intervention and to achieve robust segmentation results. In [153], an Online Random Forest (ORF) was used to efficiently update the results for interactive segmentation. The ORF avoids learning from scratch when new user interactions are added.

Recently, using deep CNNs to improve interactive segmentation has been attracting increasing attention due to CNNs' automatic feature learning and high performance. For instance, 3D U-Net [129] learns from sparsely annotated data and can be used for semi-automatic segmentation. ScribbleSup [154] also trains CNNs for semantic segmentation supervised by scribbles. DeepCut [155] combines CNNs with user-provided bounding box annotations for fetal brain and lung segmentation from fetal MR images. But these methods are not fully interactive for testing since they do not accept further interactions for refinement. In [156], a deep interactive object selection method was proposed where user-provided clicks are transformed into Euclidean distance maps and then concatenated with the input of FCNs.

2.3 Conditional Random Fields and Graph Cuts

Graphical models such as CRFs have been widely used to enhance segmentation accuracy by introducing spatial consistency for automatic and interactive methods [157, 158, 159, 160]. CRFs are a type of discriminative undirected probabilistic graphical model that can be used to encode relationships between observations and construct consistent interpretation.

Let X be a random variable over data sequences to be labeled, and Y be a random variable over corresponding label sequences. Each component y_i of Y is assumed to range over a finite label set $\mathscr{L} = \{0, 1, ..., L - 1\}$. CRFs construct a conditional model $P(Y|X)$ from paired observation and label sequences:

$$P(Y|X) = \frac{1}{Z(X)} \exp\Big(-\sum_i \psi(y_i) - \sum_{(i,j)\in\mathscr{N}} \phi(y_i, y_j) \Big) \tag{2.8}$$

where $Z(X)$ is the normalization factor known as the partition function. $\psi(y_i)$ is a

unary potential function that is defined based on observed image intensity and a likelihood function. $\phi(y_i, y_j)$ is the pairwise potential function that encourages spatial coherence by penalizing discontinuities between pixel pairs. $\mathcal{N}$ is the set of all pixel pairs and it is typically defined as neighboring pixels in the image. Inference of Y in CRFs can be implemented by maximizing the probability in Eq. (2.8). In practice this is commonly casted as a Graph Cuts problem that minimizes an energy function.

$$E(Y) = \sum_i \psi(y_i) + \sum_{(i,j)\in\mathcal{N}} \phi(y_i, y_j) \tag{2.9}$$

Graph Cuts algorithms can quickly find global optima for submodular energies [161]. If the pairwise energy $\phi(y_i, y_j)$ is positive when $y_i \neq y_j$ and zero when $y_i = y_j$, then $E(Y)$ is submodular and can be solved by Graph Cuts [162]. A graph $\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$ consists of a set of nodes (e.g., pixels, voxels or other features) and a set of directed edges that connect them. For binary image segmentation, $\mathcal{G}$ contains two additional nodes called terminals that correspond to the set of labels. The two terminals are called the source, s, and the sink, t. Normally the edges include two types: n-links that connect pairs of neighboring pixels/voxels and t-links that connect pixel-terminal pairs. Every edge is assigned some cost. The cost of n-links is derived from ϕ in Eq. (2.9), and it corresponds to a penalty for discontinuity between neighboring pixels. The cost of a t-link is derived from ψ in Eq. (2.9), and it corresponds to a penalty for assigning the corresponding label to the pixel. An s/t cut on the graph partitions the nodes into two disjoint subsets $\mathcal{S}$ and $\mathcal{T}$ such that $s \in \mathcal{S}$ and $t \in \mathcal{T}$. The cost of a cut $C = \{\mathcal{S}, \mathcal{T}\}$ is defined as the sum of the costs of "boundary" edges. Graph Cuts algorithms aim to find a cut that has the minimum cost among all cuts, i.e., minimum cut (min-cut). This problem can be solved by finding a maximum flow from the source s to the sink t. Based on the theorem of Ford and Fulkerson [163], a maximum flow from s to t saturates a set of edges in $\mathcal{G}$ dividing the nodes into two disjoint subsets $\{\mathcal{S}, \mathcal{T}\}$ corresponding to a min-cut. Therefore, min-cut and max-flow problems are equivalent.

The algorithms to solve min-cut/max-flow problems can be categorized into two

types: augmenting paths [163] and push-relabel [164]. Augmenting paths-based algorithms work by pushing flow along non-saturated paths from s to t until the maximum flow in $\mathscr{G}$ is reached. Push-relabel algorithms maintain a labeling of nodes giving a lower bound estimate on the distance to the sink along non-saturated edges, and it pushes excess flows towards nodes with smaller estimated distance to the sink. Boykov and Kolmogorov [158] extended standard augmenting path techniques to achieve improved empirical performance. This extended method builds two search trees, one from the source and the other from the sink, to detect augmenting paths. It also reuses these trees in each iteration to avoid building them from scratch. It was shown in [158] that this method significantly outperforms standard algorithms in terms of computational efficiency on typical problem instances in image restoration, stereo and object segmentation tasks.

For multi-label segmentation, Graph Cuts can be used with alpha-expansion or alpha-beta swap algorithms [165]. Let α denote one possible label, the main idea of the alpha-expansion algorithm is to successively segment all α and non-α pixels with Graph Cuts. The algorithm changes the value of α at each iteration and iterates through all the possible labels for α until it converges [165]. The alpha-beta swap algorithm successively uses Graph Cuts to segment all α pixels from pixels with a different label β, and changes the $\alpha-\beta$ combination at each iteration. The algorithm will iterate through all the possible combinations until it converges.

In order to better model long-range connections within the image, a fully connected CRF was proposed in [166] to establish pairwise potentials on all pairs of pixels in the image. To make the inference of the fully connected CRF efficient, in [127], the pairwise edge potential was defined by a linear combination of Gaussian kernels.

$$\phi(y_i, y_j) = \mu(y_i, y_j) k(\mathbf{f}_i, \mathbf{f}_j) \tag{2.10}$$

where $\mu(y_i, y_j)$ is a label compatibility function that introduces a penalty for nearby similar pixels that are assigned with different labels. Vectors $\mathbf{f}_i$ and $\mathbf{f}_j$ are feature vectors for pixel i and j in arbitrary feature space. $k(\mathbf{f}_i, \mathbf{f}_j)$ is defined in terms of the

image intensity values x_i and x_j and positions v_i and v_j:

$$k(\mathbf{f}_i,\mathbf{f}_j) = \omega_1 \exp\Big(-\frac{|v_i-v_j|^2}{2\theta_\alpha^2} - \frac{|x_i-x_j|^2}{2\theta_\beta^2}\Big) + \omega_2 \exp\Big(-\frac{|v_i-v_j|^2}{2\theta_\gamma^2}\Big) \tag{2.11}$$

where the first and second terms use a bilateral filter and a spatial filter, respectively. ω_1 and ω_2 are parameters controlling the weights of these two terms. θ_α, θ_β, θ_γ are parameters for these filters. In [127], a mean field approximation method with high dimensional filtering [167] was proposed to infer this CRF efficiently.

Parameters of CRFs in above works were manually tuned or learned by grid search with low efficiency. In [162], a maximum margin learning method was proposed to learn CRFs using Graph Cuts. Other methods including structured output SVM [168], approximate marginal inference [169] and gradient-based optimization [170] were also proposed to learn parameters in CRFs. They treat the learning of CRFs as an independent step after training classifiers.

The CRF-RNN network [171] formulated dense CRFs as RNNs so that the CNNs and CRFs can be jointly trained in an end-to-end system for segmentation. However, the pairwise potentials in [171] are limited to weighted Gaussians and not all the parameters are trainable due to the permutohedral lattice implementation [167]. In [172], a Gaussian mean field network was proposed and combined with CNNs where all the parameters were trainable. More freeform pairwise potentials for a pair of super-pixels or image patches were proposed in [173, 174], but such CRFs have a low resolution. In [175] a generic CNN-CRF model was proposed to handle arbitrary potentials for labeling body parts in depth images, but it has not yet been validated with other segmentation applications.

2.4 Co-segmentation of Multiple Images

In recent years, co-segmentation methods, which combine multiple images that provide complementary information, have been demonstrated to be able to achieve better segmentation results than methods working on a single image. For example, in [176], a general framework was proposed to use both positron emission tomography (PET) images and CT images simultaneously for tumor segmentation. This method utilizes

the strength of each imaging modality: the good contrast of PET and the high spatial resolution of CT. In [145], an algorithm for interactive co-segmentation was proposed to extract a foreground object from a group of related images. In [143], a Graph Cut approach was used to segment heart structures from multiple cardiac MR images. In [177], a coupled continuous max-flow model was proposed to jointly segment functional and structural pulmonary MR images. These works demonstrate that co-segmentation approaches yield better performance than single-image segmentation in terms of accuracy and robustness.

Considering the deformation of the same target among several different images in [178], a framework was proposed for integrating segmentation and registration through active contours that can simultaneously segment and register features from multiple images. In [179], a general framework was introduced for co-segmentation and registration of the kidney from contrast enhanced ultrasound images and traditional ultrasound images. In [180], a level set-based framework was proposed for simultaneous registration, segmentation and shape interpolation from misaligned images with large inter-slice spacing.

2.5 Basics of Deep Convolutional Neural Networks

CNN is a special type of neural networks. It is designed to better utilize spatial information by taking 2D or 3D images as input with three mechanisms: local receptive field, shared weights and pooling [181]. An illustration of these mechanisms is shown in Fig. 2.2. Let $A_j^{(l)}$ be the j-th feature map at l-th layer. A convolution layer uses a learnable kernel $k_{ij}^{(l)}$ to represent a connection between $A_i^{(l-1)}$ and $A_j^{(l)}$.

$$A_j^{(l)} = f\Big(\sum_{i=1}^{M^{(l-1)}} A_i^{(l-1)} * k_{ij}^{(l)} + b_j^{(l)}\Big) \tag{2.12}$$

where $M^{(l-1)}$ is the number of feature maps in layer $l-1$ and $*$ is a convolution operator. $b_j^{(l)}$ is a bias parameter and $f(\cdot)$ is a non-linear activation function, e.g., sigmoid, tanh, rectified linear unit (ReLU) and leaky ReLU.

Fig. 2.3 shows four different non-linear activation functions. The sigmoid non-

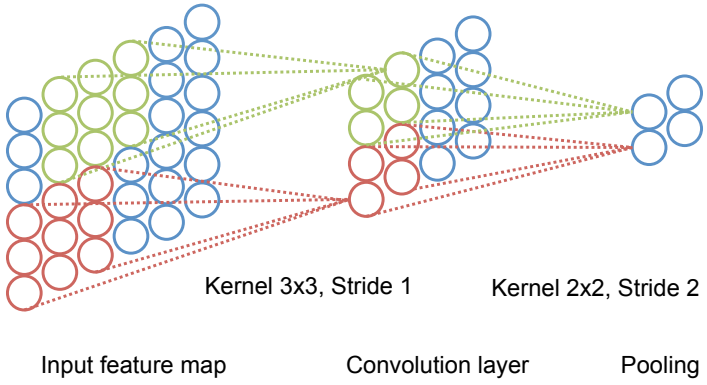


Figure 2.2: An illustration of mechanisms of convolutional neural networks: local receptive field, shared weights and pooling.

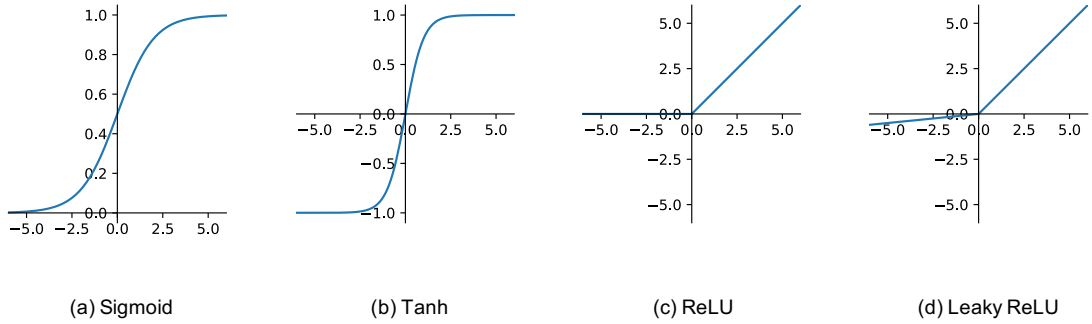


Figure 2.3: Four different non-linear activation functions.

linearity has the mathematical form $\sigma(x) = 1/(1+e^{-x})$. It maps a real-valued number to a value in the range between 0 and 1. A sigmoid function is monotonic and differentiable, with a non-negative first derivative which is bell shaped. A drawback of the sigmoid neuron is that the neuron's activation saturates at either tail of 0 or 1, which makes the gradient at these regions almost zero, i.e., vanishing gradient problems. In addition, sigmoid outputs are not zero-centered. This can introduce undesirable zig-zagging dynamics in the gradient updates for the weights. The tanh function squashes a real-valued number to the range (-1,1), with the mathematical form $\tanh(x) = 2\sigma(2x) - 1$. This function has a similar shape to that of sigmoid, but its outputs are zero-centered.

The ReLU function became popular in recent years [182]. It is defined as $f(x) = \max(0,x)$. Despite the simplicity, ReLU has several advantages. It was found to largely accelerate the convergence of training of deep neural networks compared to

the sigmoid/tanh functions, due to its linear and non-saturating form. It can be simply implemented by thresholding an array of activations at zero, without expensive exponential operations. A drawback of ReLU is that it can lead some neurons to "die" during training, where the neuron becomes stuck in a perpetually inactive state, i.e., the "dying ReLU" problem.

Leaky ReLU allows a small, positive gradient when the unit is not active. It is defined as $f(x) = \max(x, ax)$ where $a \leq 1$ is a parameter. Leaky ReLU is one attempt to fix the "dying ReLU" problem. Instead of outputting zero when $x < 0$, it will instead have a small negative slope a. He et al. [182] made a of each neuron trainable, and proposed a Parametric Rectified Linear Unit (PReLU) that generalizes the traditional rectified unit.

For image classification and segmentation tasks, the softmax function is often used in the final layer of a neural network, so that a "probability" corresponding to each class or discrete label can be obtained. Softmax is a generalization of the logistic function that "squashes" a K-dimensional vector $\mathbf{z}$ of arbitrary real values to a K-dimensional vector $\sigma(\mathbf{z})$ of real values. Each entry of the output is in the range (0, 1), and the sum of all the entries is 1. The function is defined as:

$$\sigma(\mathbf{z})_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \quad \text{for} j = 1, 2, ..., K \tag{2.13}$$

Before the training process, the parameters of a neural network need to be initialized. The initial values of convolution parameters are usually set to random numbers that are very close to zero with a variance. He et al. [182] concluded that the variance of ReLU neurons in the network should be $2.0/n$, where n is the number of inputs.

The learning of a neural network is a process to minimize an empirical loss function based on training samples. Let g_i be the ground truth for sample i, and the corresponding prediction result by a neural network is y_i. The loss function for training is:

$$L = \frac{1}{N} \sum_{i=1}^{N} L_i(y_i, g_i) \tag{2.14}$$

where N is the number of training samples. $L_i(y_i, g_i)$ represents the loss between y_i and g_i. For regression problems, L_i is usually defined as a squared difference function $L_i(y_i, g_i) = (y_i - g_i)^2$. For classification problems, the logarithmic loss function is often used. It measures the performance of a classification model which outputs a probability for each class. In this case, the ground truth is represented as a one-hot vector $\mathbf{g}_i$. Assume the number of classes is K, then it equals to the length of $\mathbf{g}_i$. For $k = 1, 2, ..., K$, $\mathbf{g}_{ik}$=1 if $k = g_i$ and 0 otherwise. Let $\mathbf{p}_i$ be the vector of probability corresponding to each class predicted by the network. The logarithmic loss is defined as:

$$L_i(\mathbf{g}_i, \mathbf{p}_i) = -\sum_{k=1}^{K} \mathbf{g}_{ik} \log(\mathbf{p}_{ik}) \tag{2.15}$$

For binary classification problems where $K = 2$, Eq. (2.15) becomes the cross entropy loss function. Let p_i represent the probability of i being labeled as 1. The cross entropy loss is:

$$L_i(g_i, p_i) = -\big(g_i \log(p_i) + (1 - g_i) \log(1 - p_i)\big) \tag{2.16}$$

Let ω represent the parameters of the neural network. With the loss function in Eq. (2.14), the parameters of a deep neural network can be iteratively updated through gradient-based optimization methods. The most commonly used optimization method is Stochastic Gradient Descent (SGD). In SGD, the true gradient of $L(\omega)$ is approximated by the gradient at a single training sample. The simplest update of ω has the form (vanilla SGD):

$$\omega_t = \omega_{t-1} - \eta \nabla L_i(\omega) \tag{2.17}$$

where t is the step index and η is the learning rate. As the algorithm sweeps through the training set, it performs the parameter update for each training sample. The training set can be traversed for several epochs until the algorithm converges.

Since the gradient at a single sample is a noisy approximation of the true gradient,

training with vanilla SGD may cause the model parameters to jump around. To alleviate this problem, parameters can be updated using "momentum", which helps better converge rates on deep neural networks.

$$v_t = \beta v_{t-1} + \eta \nabla L_i(\omega) \tag{2.18}$$

$$\omega_t = \omega_{t-1} - v_t \tag{2.19}$$

where β is a parameter for the momentum. Using momentum corresponds to calculating exponentially weighted averages of the gradients at different iteration steps, and it provides a better estimate of the true gradient. Therefore, it might work better than vanilla SGD. In addition, in each step the gradient can be computed against a small set of training samples, i.e., mini-batch. It usually results in smoother convergence. Several other variants of SGD have also been proposed for better convergence of training, e.g, using adaptive sub-gradient (AdaGrad) [183], adaptive learning rate (RMSProp) [184], and adaptive moment estimation (Adam) [185].

Ioffe et al. [186] proposed batch normalization to accelerate deep network training by explicitly forcing the activations throughout a network to take on a unit Gaussian distribution. Batch normalization allows the use of higher learning rates and makes the training less sensitive to initializations.

In order to prevent the network from over-fitting training samples, some regularization methods can be used. L2 regularization penalizes the squared magnitude of all parameters, whereas L1 regularization penalizes their absolute magnitudes, leading to the weight vectors becoming sparse during optimization. Max-norm constraint enforces an absolute upper bound on the magnitude of the weight vector for neurons and uses projected gradient descent to enforce the constraint. It has been shown that batch normalization also acts as a regularization method [186].

Srivastava et al. [187] proposed Dropout as an effective and simple way to prevent neural networks from over-fitting. The core idea is to randomly drop units along with their connections from the neural network during training. It can be interpreted as extracting a random sub-network from the full neural network, and only updating the parameters of the sub-network. At test time, there is no dropout applied, which can

be interpreted as evaluating an averaged prediction across the ensemble of all sub-networks. Dropout was found to improve the performance of neural networks in a wide variety of applications such as object classification, speech recognition and biological data analysis [187].

2.6 Segmentation of Fetal MR Images

This thesis deals with the task of placenta segmentation from fetal MR images. However, only a few works have been reported for this task. For fetal MR image segmentation, most previous works have focused on the fetal brain [188, 189, 190]. Thus, I give a review of not only placenta segmentation, but also segmentation of the fetus or other fetal organs and segmentation of objects from motion-corrupted volumes that are related to the work in this thesis.

There are mainly two categories of methods for segmentation of fetal organs from fetal MR images. The first one is to segment the target organ directly from a single motion-corrupted volume. In [188], a shape prior model was used to extract head structures from 2D fetal MR images, and the results were used to guide a 3D segmentation. In [190], an automatic way was proposed to segment the fetal brain slice-by-slice by localizing the brain area first and then using RFs for patch classification. The segmented results are used to construct a high-resolution volume. In [191], an auto-context CNN was introduced to extract fetal brains from fetal MR images for entire brain analysis. In [192], median filtering was used as pre-processing to attenuate motion artifacts between slices, and a method based on RFs and steerable features was proposed to localize and segment the heart, the lungs and the liver of the fetus. In [193], a graph-based method was used for whole body segmentation from fetal MR images.

The second category first uses multiple volumes to reconstruct a single high-resolution volume and then segments the reconstructed image. In [71, 189, 194], a registration between many 2D slices and a 3D volume was used to reconstruct a high-resolution volume from multiple motion-corrupted fetal MR images, and after that the fetal brain was segmented by a probabilistic atlas-based method.

In terms of placenta segmentation, in [63], a random walker algorithm was used

to interactively segment the placenta from 3D ultrasound. In [195], a 3D multi-scale CNN was adopted to automatically segment the placenta from motion-corrupted MR images. However, accurate and robust results are hard to achieve due to the poor quality of 3D fetal MR images and the large variation of the placenta among pregnant women. This thesis focuses on developing interactive methods for accurate segmentation of the placenta from fetal MR images, including 2D slices, 3D volumes and multiple volumes of the same patient (4D).

Chapter 3

Dynamically Balanced Online Random Forests for Interactive Scribble-based Segmentation

3.1 Introduction

The method and results presented in this chapter have been published as a conference paper in MICCAI 2016 [74].

In this chapter, I propose an Online Random Forest (ORF)-based interactive method for placenta segmentation from 2D fetal MR slices. The goal is to extract the placenta from the background, and this is a binary segmentation problem, i.e., pixel-wise binary classification problem.

For binary classification problems, let P and N represent the set of positive samples and negative samples. The whole training set $S = P \cup N$ is balanced if the sizes of P and N are close, i.e., $|P| \approx |N|$. When $|P|$ is considerably smaller or larger than $|N|$, then S becomes imbalanced. In this chapter, I define the imbalance ratio of S as $\gamma = |N|/|P|$, i.e., the ratio between negative sample number and positive sample number. For a balanced training set, γ is close to 1.0.

In the context of scribble and learning-based segmentation, the user-provided scribbles for the foreground and background are used to train a classifier. The positive samples are pixels in the foreground scribbles, and the negative samples are pix-

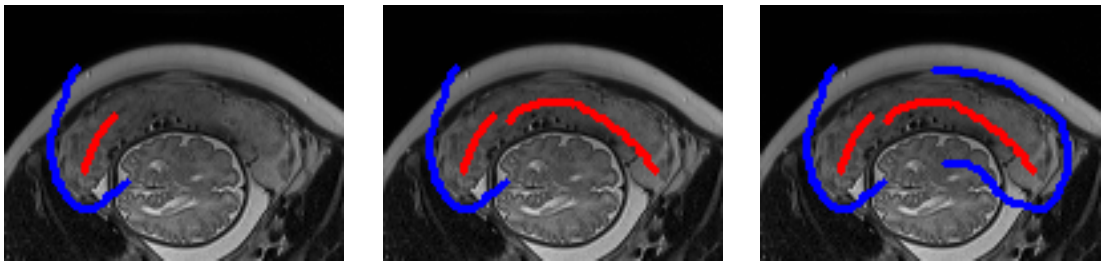
els in the background scribbles. During interactive segmentation, the scribbles for the foreground and background are usually imbalanced. Fig. 3.1(a) shows an example of scribbles at the beginning of an interactive segmentation process, where the user draws more background scribbles than foreground scribbles, and the imbalance ratio γ is 2.0. Traditional RFs deal with imbalanced training data with the assumption that the imbalance ratio is fixed during the learning process [196, 197]. However, when the user draws scribbles gradually, the imbalance ratio between the foreground and background scribbles can change. For example, γ becomes 0.7 and 1.8 in Fig. 3.1(b) and Fig. 3.1(c) respectively when the user provides more scribbles. Although ORFs have been used for interactive binary segmentation in [148, 153], they have limited ability to learn from imbalanced training data with a changing imbalance ratio. Failing to deal with this problem could limit the performance of RFs for interactive segmentation.

To overcome this problem, I propose a generic Dynamically Balanced Online Random Forest (DyBa ORF) to deal with incremental and imbalanced training data with a changing imbalance ratio.

I validate DyBa ORF with two different applications: learning-based interactive segmentation of the placenta from fetal MR images and adult lungs from chest radiographs. In these applications, the segmentation tasks are challenging due to low contrast between the target and the background as well as inhomogeneous appearances. This motivates the use of high-level features combined with DyBa ORF-based learning rather than a traditional GMM, which is often used to model low dimensional features and not well suited to online learning. The experiments demonstrate DyBa ORF outperforms traditional ORFs in these two applications, with its ability to achieve comparable accuracy and higher efficiency compared with its offline counterpart.

3.2 Method

The workflow of using DyBa ORF for interactive placenta segmentation is shown in Figure 3.2. In the proposed method, the user draws scribbles to label some pixels to be the foreground and the background, respectively. These pixels are used as training data of DyBa ORF. After training, the DyBa ORF predicts the label of the remaining



(a) Initial scribbles imbalance ratio $\gamma = 2.0$ (b) More foreground scribbles imbalance ratio $\gamma = 0.7$ (c) More background scribbles imbalance ratio $\gamma = 1.8$

Figure 3.1: An illustration of imbalanced training data with a changing imbalance ratio for interactive segmentation. Foreground scribbles (red) and background scribbles (blue) are imbalanced, and the ratio of background scribbles to foreground scribbles changes from (a) to (c) when the user draws scribbles additionally.

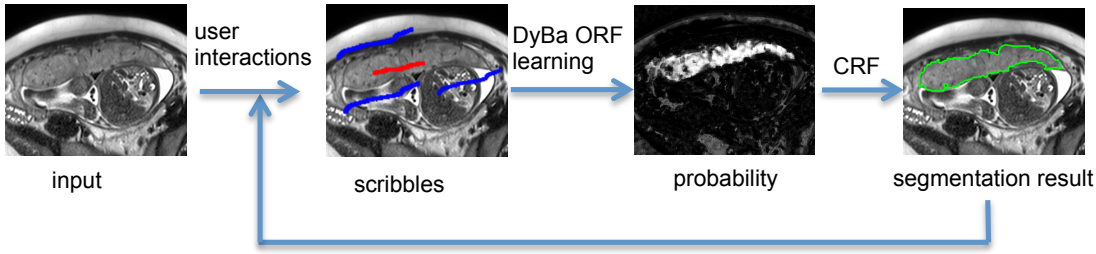


Figure 3.2: Workflow of using DyBa ORF for interactive placenta segmentation.

pixels and gives a probability of each pixel belonging to the foreground. To get a more spatially consistent result, a CRF is used to reduce noise in the segmentation result. The user may give some additional scribbles to refine the initial result. The new scribbles are added to the existing training set of DyBa ORF, which is dynamically updated on the fly without learning from scratch. The updated DyBa ORF gives a new probability prediction, and then the CRF is applied again. These steps are iterated until the user accepts the segmentation result.

3.2.1 Feature Extraction

For each pixel, features are extracted from a local Region of Interest (ROI) centered on it. In each ROI, the extracted features include gray level features, texture features and wavelet features that have shown effective in previous works [198]. The gray level features are based on mean and standard deviation of intensity in a local patch. Texture features are acquired by gray level co-occurrence matrix (GLCM) [199]. The co-occurrence probability is a second-order method for generating texture features. It

Table 3.1: Gray level co-occurrence texture statistics.

Maximum probability	$\max\{P_{ij}\}$ for all (i,j)
Uniformity	$\sum P_{ij}^2$
Entropy	$\sum P_{ij} \log P_{ij}$
Dissimilarity	$\sum P_{ij}\lvert i-j\rvert$
Contrast	$\sum P_{ij}(i-j)^2$
Correlation	$\sum \frac{(i-\mu_x)(j-\mu_y)P_{ij}}{\sigma_x \sigma_y}$
Inverse difference	$\sum \frac{P_{ij}}{1+\lvert i-j\rvert}$
Inverse difference moment	$\sum \frac{P_{ij}}{1+(i-j)^2}$

represents the conditional joint probability of all pairwise combination of gray levels in a spatial window given an inter-pixel offset (δ) [200]. The probability measure is defined as:

$$P_{ij} = \frac{N_{ij}}{\sum_{i}^{G}\sum_{j}^{G} N_{ij}} \tag{3.1}$$

where N_{ij} represents the number of occurrences of quantized gray levels i and j within the given window with a certain offset δ. G is the total number of quantized gray levels. The sum in the denominator represents the total number of gray level pairs (i,j) within the window. Statistics based on the co-occurrence probabilities are used to generate texture features. Commonly used statistics are shift-invariant, such as uniformity, entropy and contrast. These statistics as listed in Table 3.1.

Wavelet features are based on Discrete Wavelet Transform (DWT) [201]. It captures both frequency and location information, which is a key advantage over Fourier transforms. DWT decomposes a signal into a set of mutually orthogonal wavelet basis functions that are spatially localized. 2D DWT decomposes an image into wavelet coefficients with horizontal and vertical high/low-pass filters. The outputs give the detail coefficients from the high-pass filter and approximation coefficients from the low-pass filter [202]. One of the most common basis functions is the Haar wavelet [203]. Fig. 3.3 illustrates examples of 2D Haar wavelet decomposition. For each level of DWT, there are four bounds: LL, LH, HL and HH, where L stands for low-pass fil-

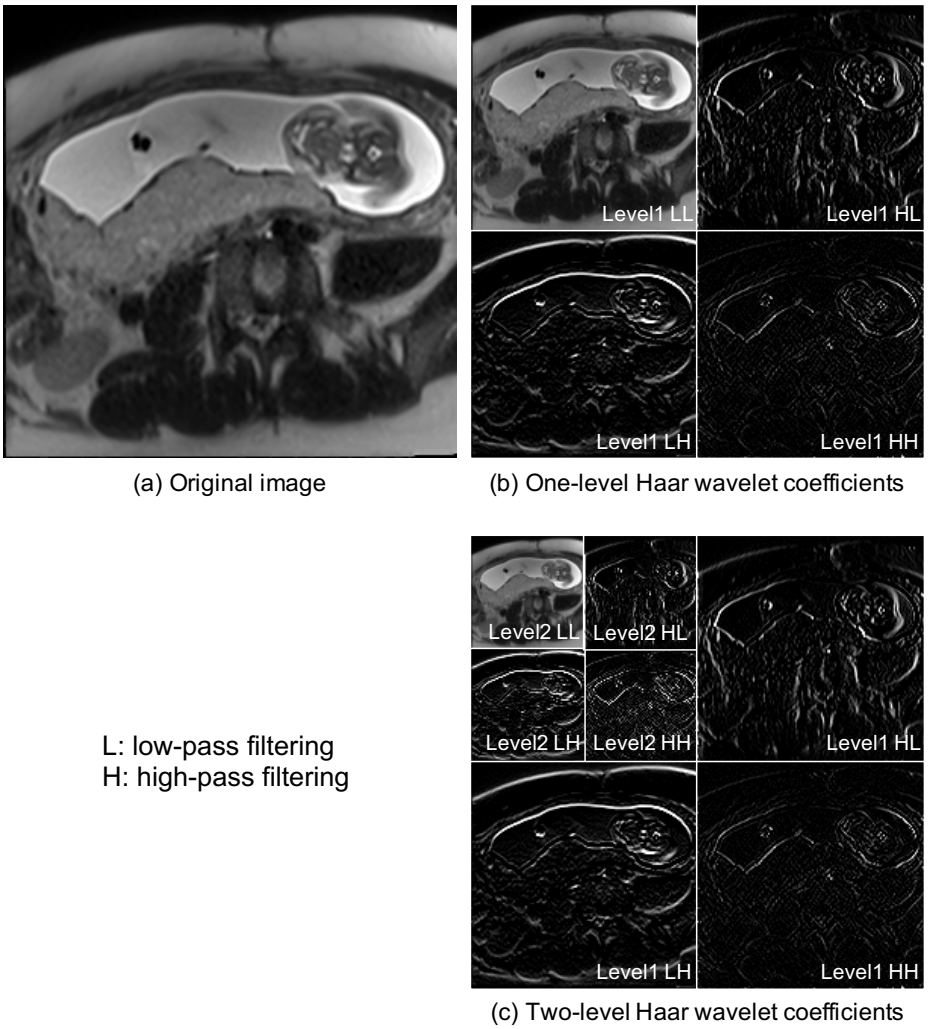


Figure 3.3: Examples of Haar wavelet decomposition.

tering and H stands for high-pass filtering. The LL band contains approximation coefficients and it corresponds roughly to a down-sampled version of the original image. The LH, HL and HH bands contain the horizontal, vertical and diagonal detail coefficients, respectively. Statistical measurements such as mean and standard deviation of the Haar wavelet coefficients in a local patch are calculated as wavelet features. The extracted features based on gray level, texture and wavelet are used to train a classifier for segmentation in the following section.

3.2.2 Dynamically Balanced Online Random Forests

3.2.2.1 Traditional ORFs and their Limitations

Random Forest is a widely used machine learning method [112] that has shown to be efficient with competitive performance. A Random Forest is a set of N binary decision trees with split nodes and leaf nodes. A split node executes a binary split function to propagate a sample to its left or right child, and a leaf node stores all the training samples that have been propagated to it. The split function for the j-th split node is:

$$f_\theta^j(\mathbf{x}_i) \in \{0,1\} \tag{3.2}$$

where $\mathbf{x}_i$ is a training or testing sample for the split node and θ is the parameter of the split function. $\mathbf{x}_i$ is sent to the left child if $f_\phi^j(\mathbf{x}_i) = 0$ or the right child if $f_\phi^j(\mathbf{x}_i) = 1$.

During the construction of a tree, a node in the tree splits when the number of samples in that node is higher than a threshold value and a split criterion is satisfied. θ for each split function is optimized by maximizing the split criterion of the corresponding split node. The split criterion is often based on Information Gain, Gini Index or Variance Reduction [204]. Assume a split node s has a left child node s_l and a right child node s_r, the Information Gain is defined as:

$$IG(s) = H(s) - \Big(\frac{N_l}{N_l+N_r}H(s_l) + \frac{N_r}{N_l+N_r}H(s_r)\Big) \tag{3.3}$$

where N_l and N_r are the number of samples in s_l and s_r, respectively. $H(s)$ is the entropy of a node: $H(s) = \sum_{j=1}^{J} -p_j log_2 p_j$ where p_j is the probability of class j in node s. Gini Index is defined as:

$$GI(s) = G(s) - \Big(\frac{N_l}{N_l+N_r}G(s_l) + \frac{N_r}{N_l+N_r}G(s_r)\Big) \tag{3.4}$$

where $G(s)$ is the Gini impurity: $G(s) = \sum_{j=1}^{J} p_j(1-p_j)$. Previous studies found that Information Gain and Gini Index lead to very close performance and it is difficult to conclude which one of them is better [205]. Variance Reduction is often employed for regression problems where the target variable is continuous.

In the testing stage, the distribution of class labels stored in a leaf is used for prediction. Each test sample $\mathbf{x}_i$ in a slice X is propagated through all trees in the forest. For the nth tree, a posterior probability $p_n(y_i|\mathbf{x}_i,X)$ is obtained from the leaf that the test sample falls into, where y_i is the label of $\mathbf{x}_i$. The final posterior is achieved as the average across all the N trees.

$$p(y_i|\mathbf{x}_i,X) = \frac{1}{N}\sum_{n=1}^{N} p_n(y_i|\mathbf{x}_i,X) \tag{3.5}$$

To overcome over-fitting, the training set of each tree is obtained by randomly resampling (a.k.a. bootstrap aggregating, or Bagging) the original training set for the forest. In [206], the traditional RFs were extended to Online Random Forests (ORF) to deal with online learning problems. The ORFs [206] use online Bagging that models the sequential arrival of data as a Poisson distribution Pois(λ) with a rate of λ. Each tree is updated on each new training sample k times where $k \sim$Pois(λ) and the expectation of k is λ. To deal with offline learning with imbalanced data, weighting samples and re-sampling the training set were proposed in [197]. For online learning with imbalanced data, different values of λ for Poisson distributions were used in [153] for different classes based on the imbalance ratio. After receiving new training samples that lead to a new imbalance ratio, this method samples the new data with a rate based on the new imbalance ratio to grow existing trees, but does not update the set of existing sampled training data that has been sampled with a rate based on the old imbalance ratio. Thus, it fails to be truly adaptive to imbalance ratio changes.

3.2.2.2 Dynamically Balanced Online Bagging

For the sake of simplicity, this chapter focuses on a binary classification problem, and the proposed method can be easily extended to multi-class problems. Suppose at an initial stage of online learning, the training data for the forests is represented by a tuple $S_0(P_0,\ N_0)$ where P_0 is a set of positive samples and N_0 is a set of negative samples. The initial imbalance ratio is defined as $\gamma_0 = |N_0|/|P_0|$. There are three options to deal with imbalanced data: weighting samples, up-sampling the minority class, and down-sampling the majority class [197]. This chapter chooses to down-sample the

majority class for efficiency. Suppose Pois(λ) is used to resample the minority class, then Pois(λ_{p0}) and Pois(λ_{n0}) are used to resample P_0 and N_0, respectively:

$$\lambda_{p0} = \begin{cases} \lambda, & \text{if } \gamma_0 \geq 1.0 \\ \lambda\gamma_0, & \text{otherwise} \end{cases}; \qquad \lambda_{n0} = \begin{cases} \lambda/\gamma_0, & \text{if } \gamma_0 \geq 1.0 \\ \lambda, & \text{otherwise} \end{cases} \tag{3.6}$$

Thus, each sample in P_0 is expected to be sampled λ_{p0} times, and each sample in N_0 is expected to be sampled λ_{n0} times. The sampled training set for a certain tree is denoted as $S_0^*(P_0^*,\ N_0^*)$, where P_0^* and N_0^* are sampled from P_0 and N_0, respectively. $|P_0^*|$ has an expectation of $\lambda_{p0}|P_0|$, and $|N_0^*|$ has an expectation of $\lambda_{n0}|N_0| = \lambda_{n0}\gamma_0|P_0| = \lambda_{p0}|P_0|$. Therefore, the sampled training set S_0^* is balanced and it is used to construct the tree.

When a set of new training samples $S_\dagger(P_\dagger,\ N_\dagger)$ arrive, $S_\dagger$ is added into S_0. A merged training set $S_1(P_1,\ N_1)$ is obtained, where $P_1 = P_0 \cup P_\dagger$ and $N_1 = N_0 \cup N_\dagger$. The new imbalance ratio is $\gamma_1 = |N_1|/|P_1|$. In an offline situation, Pois(λ_{p1}) and Pois(λ_{n1}) should be used to sample P_1 (obtaining P_1^*) and N_1 (obtaining N_1^*), respectively, where λ_{p1} and λ_{n1} are defined based on γ_1 and λ in the same way as shown in Eq. (3.6). For online learning, instead of sampling P_1 and N_1 to get P_1^* and N_1^* from scratch, the proposed method dynamically updates P_0^* and N_0^* to obtain P_1^* and N_1^*. It generates an Add Set A and a Remove Set R from both $S_\dagger$ and S_0 based on the imbalance ratio change in the following way:

For the newly arrived sample set $S_\dagger$, a standard balanced sampling procedure is applied by using Pois(λ_{p1}) and Pois(λ_{n1}) to sample $P_\dagger$ (obtaining $P_\dagger^*$) and $N_\dagger$ (obtaining $N_\dagger^*$), respectively. This results in a sampled subset $S_\dagger^*(P_\dagger^*,\ N_\dagger^*)$, and $S_\dagger^*$ is added to A.

For the old sample set S_0, its positive subset P_0 and negative subset N_0 are dealt with respectively. The expected sampling rate for P_0 should be the same as that for $P_\dagger$, and the expected sampling rate for N_0 should be the same as that for $N_\dagger$. For N_0, with the new Poisson distribution Pois(λ_{n1}), each sample in N_0 is expected to be sampled λ_{n1} times. The expected difference of sampling rate between before and after $N_\dagger$ arrives is $\delta_n = \lambda_{n1} - \lambda_{n0}$. If $\delta_n > 0$, it means after the new data arrive, more negative data should be sampled from N_0 in order to keep the same sampling rate Pois(λ_{n1}) as used for $N_\dagger$. N_0 is additionally sampled with Pois(δ_n) to get an Add Set A_n that is added

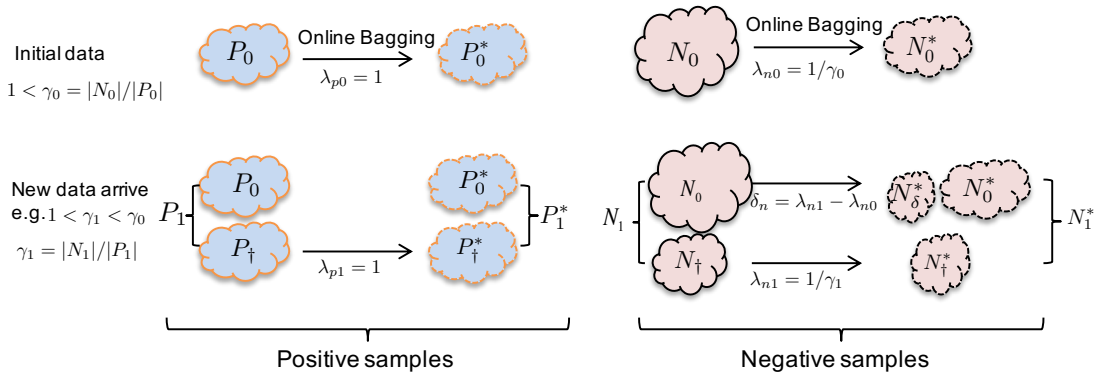


Figure 3.4: An example of dynamically balanced Bagging. The initial training data set has more negative samples (N_0) than positive samples (P_0). P_0 and N_0 are resampled with Pois($\lambda_{p0} = 1$) and Pois($\lambda_{n0} = |P_0|/|N_0|$), respectively. Thus, the sampled result P_0^* and N_0^* are balanced. When new training data (positive $P_\dagger$ and negative $N_\dagger$) arrive, $P_\dagger$ and $N_\dagger$ are resampled with Pois($\lambda_{p1} = 1$) and Pois($\lambda_{n1} = |P_1|/|N_1|$), respectively, where $P_1 = P_0 \cup P_\dagger$ and $N_1 = N_0 \cup N_\dagger$. In the case of $\delta_n = \lambda_{n1} - \lambda_{n0} > 0$, to make N_0 and $N_\dagger$ be sampled with the same parameter (λ_{n1}), more samples are sampled from N_0 with Pois(δ_n), obtaining N_δ^*. Thus, the new resampled positive set $P_1^* = P_0^* \cup P_\dagger^*$ and negative set $N_1^* = N_0^* \cup N_\delta^* \cup N_\dagger^*$ are balanced.

to A. An example of this situation is shown in Fig. 3.4. If $\delta_n < 0$, it means after the new data arrive, fewer positive samples from N_0^* are needed to keep the same sampling rate Pois(λ_{n1}) as used for $N_\dagger$. A random number $r \sim$Pois($|\delta_n| \times |N_0|$) is generated, and min(r, $|N_0^*|$) samples are sampled from N_0^* to obtain a Remove Set R_n that is added to R.

The same steps are used to deal with S_0's positive subset P_0, so that either an Add Set A_p or a Remove Set R_p is obtained. Thus, the whole Add Set is $A = S_\dagger^* \cup A_p \cup A_n$, and the whole Remove Set is $R = R_p \cup R_n$. To get the updated training sample set S_1^* for a tree on the fly, R is removed from S_0^* and A is added to it: S_1^*=$(S_0^* - R) \cup A$. Thanks to the way R and A are generated, S_1^* is balanced and adapted to the new imbalance ratio γ_1.

3.2.2.3 Tree Growing and Shrinking

Instead of reconstructing trees from scratch, the Remove Set R and Add Set A are used to update an existing tree that has been constructed based on S_0^*, to make the updated tree adapted to the imbalance ratio change. Each sample in R and A is propagated from the root to a leaf. Assume a subset R_l of R and a sub set A_l of A fall into one certain

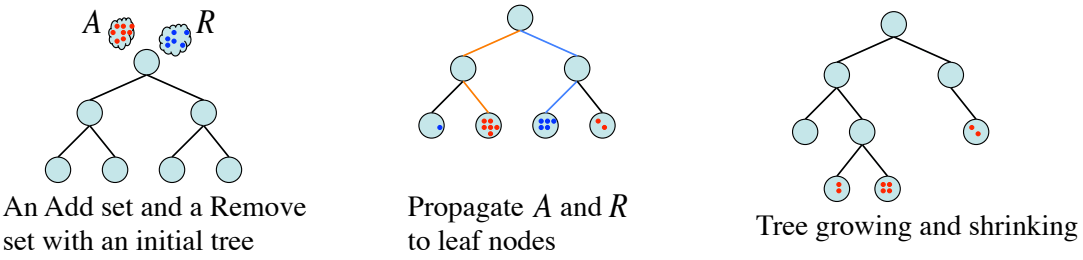


Figure 3.5: An example of tree update based on an Add set and a Remove set.

leaf l with an existing sample set S_{l_old}, the sample set of l is updated as $S_{l\_new} = (S_{l\_old} - R_l) \cup A_l$. Then, tree growing or shrinking is implemented on l based on S_{l_new}. If $|S_{l\_new}| > 0$, a split test is executed for l and its children are created (i.e. growing) if applicable based on the same split rules as used in the tree constructing stage [112]. If $|S_{l\_new}| = 0$, l is deleted (i.e. shrinking). Its parent merges the left and right child and becomes a leaf. The parent of a deleted leaf is tested for growing or shrinking again if applicable. An example of tree growing and shrinking is shown in Fig. 3.5.

3.2.3 Conditional Random Fields

In the testing stage of ORF, the posterior probability for each pixel is obtained independently. This leads the result to be sensitive to noise and lack spatial consistency. To address this problem and infer the label set for all the pixels in a slice, a CRF is used for global spatial regularization. The label set Y of a slice is determined by minimizing the following energy function.

$$E(Y) = \sum_{i \in X} \psi(y_i|\mathbf{x}_i, X) + \lambda_1 \sum_{\{i,j\} \in \mathcal{N}_1} \phi(y_i, y_j|X) \tag{3.7}$$

$$\psi(y_i|\mathbf{x}_i, X) = -\log p(y_i|\mathbf{x}_i, X) \tag{3.8}$$

$$\phi(y_i, y_j|X) = B_{i,j} \cdot \delta_{i,j} \tag{3.9}$$

where λ_1 is a coefficient to adjust the weight between two potentials. The unary potential $\psi(y_i|\mathbf{x}_i, X)$ measures the cost for assigning a class label y_i to pixel i in a slice X,

and probability p comes from the output of ORF. $\mathcal{N}_1$ is the set of all unordered pairs of $\{i, j\}$ of neighboring pixels in the slice. The pairwise potential $\phi(y_i, y_j|X)$ is defined as a contrast sensitive Potts model. $\delta_{i,j}$ equals to 1 if $y_i \neq y_j$ and 0 otherwise. $B_{i,j}$ measures the energy due to the difference in intensity between two neighboring pixels. This chapter uses a typical definition of $B_{i,j}$ proposed by Boykov et al. [133]:

$$B_{i,j} = \frac{1}{dist(i,j)} \cdot exp\left(-\frac{(x_i - x_j)^2}{2\sigma_1^2}\right) \tag{3.10}$$

where x_i and x_j denote the intensity of pixel i and j respectively. Here intensity values are used rather than feature values for efficiency. $dist(i, j)$ is the spatial distance between two neighboring pixels, and σ_1 controls the sensitivity of difference between x_i and x_j. The energy minimization in Eq. (3.7) is solved by a max-flow algorithm [133].

3.3 Experiments and Results

DyBa ORF was compared with three counterparts: 1) SP ORF: a traditional ORF with a single Poisson distribution Pois(λ) for both foreground and background class without considering the imbalance, 2) MP ORF: a traditional ORF [153] with multiple Poisson distributions based on Eq. (3.6). It uses two fixed values of λ_{p0} and λ_{n0} for the foreground and background class respectively to address the data imbalance problem, but does not deal with the change of imbalance ratio, and 3) OffBa RF: an offline counterpart that uses Poisson distributions based on λ_{p1} and λ_{n1} and learns from scratch when new data arrive. The parameter settings were: λ = 1.0, λ_1 = 5.0, σ_1 = 4.8, tree number 50, the maximal tree depth 20, the minimal sample number for split 6. The ROI size for feature extraction was 9×9. The code was implemented in C++ and made publicly available[1].

3.3.1 Validation of DyBa ORF

In the first part of experiments, DyBa ORF was validated as an online learning algorithm with four of the UCI data sets[2] that are widely used in machine learning community [207, 208]: QSAR biodegradation, Musk (Version 1), Cardiotocography and

[1] https://github.com/gift-surg/DyBaORF
[2] http://archive.ics.uci.edu/ml/datasets.html

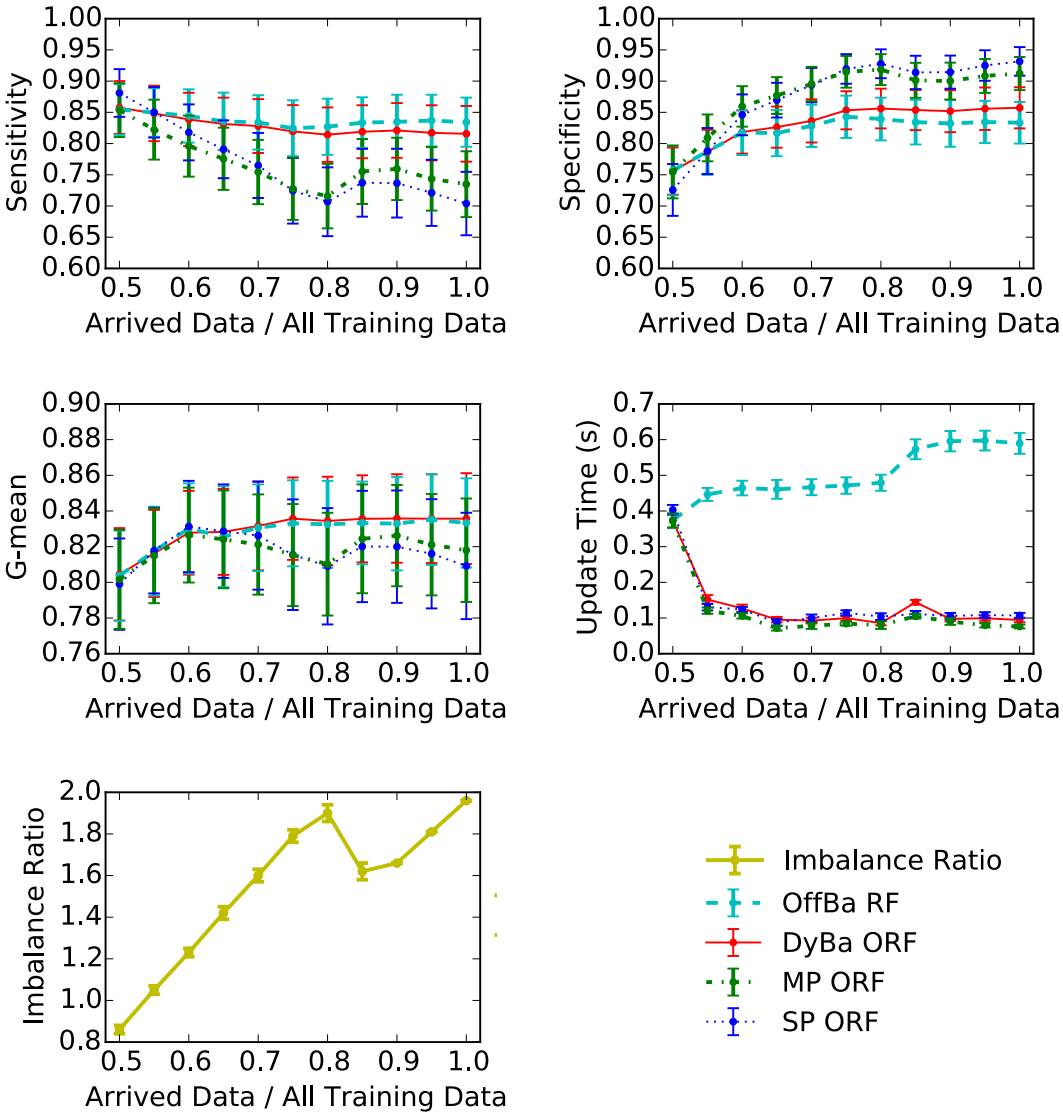


Figure 3.6: Performance of DyBa ORF and counterparts on UCI QSAR biodegradation data set. Training data were gradually obtained from 50% to 100%.

Wine. The positive class labels for them are “RB”,“1”,“8” and “8”, respectively. Each of these data sets has an imbalance between the positive and negative class. A Monte Carlo cross-validation with 100 repetition times was used. In each repetition, 20% positive samples and 20% negative samples were randomly selected to constitute test data. The remaining 80% samples were used as training data T in an online manner. The initial training set S_0 contained the first 50% of T and it was gradually enlarged by the second 50% of T, with 5% of T arriving each time in the same order as they appeared in T.

Table 3.2: G-mean of DyBa ORF and counterparts on four UCI data sets after 100% training data arrived during online learning. The bold font shows values that are not significantly different from the corresponding results of OffBa RF(p-value of Student's t-test$>$0.05). The G-mean of SP ORF on Wine is zero due to classifying all the samples into the negative class.

Data Set	Biodegradation	Musk(version1)	Cardiotocography	Wine
OffBa RF	83.33±2.50	82.78±4.44	97.06±1.69	76.14±3.34
DyBa ORF	**83.57±2.55**	**83.06±4.08**	**97.09±1.56**	**76.50±3.91**
MP ORF	81.80±2.90	73.83±6.95	95.52±1.16	74.99±4.39
SP ORF	80.92±2.98	81.65±4.93	87.59±5.41	0.00±0.00

For quantitative evaluations, I measured classification sensitivity, specificity and the update time of the forest when new data arrive. In addition, I used G-mean, which is a more suitable evaluation measure for imbalanced data than the overall accuracy and has been used in previous works [209, 207].

$$\text{G-mean} = \sqrt{\text{sensitivity} \times \text{specificity}} \tag{3.11}$$

Table 3.2 shows the final G-mean on all the four datasets after 100% T arrived. The performances on the QSAR biodegradation data set are presented in Fig. 3.6, which shows a decreasing sensitivity and increasing specificity for SP ORF and MP ORF. In contrast, OffBa RF keeps high sensitivity and G-mean when the imbalance ratio increases. The sensitivity and specificity of DyBa ORF are close to those of OffBa RF, but DyBa ORF takes much less time to update the forest when new data arrive.

3.3.2 Interactive Segmentation of the Placenta and Adult Lungs

In this experiment, DyBa ORF was applied to two different 2D segmentation tasks: placenta segmentation from fetal MR images and adult lung segmentation from chest radiographs. Stacks of MR images from 16 patients in the second trimester were acquired with SSFSE. The images have a slice dimension of 512×448 and a pixel spacing of 0.7422mm×0.7422mm. A slice in the middle of each placenta was used, with the ground truth manually delineated by a Radiologist. Lung images and ground truth[3] were downloaded from the JSRT Database[4]. Data from the first 20 normal patients

[3] http://www.isi.uu.nl/Research/Databases/SCR/
[4] http://www.jsrt.or.jp/jsrt-db/eng.php

were used in this study. The image size was 2048×2048, and the pixel spacing was $0.175\text{mm} \times 0.175\text{mm}$. At the start of segmentation, the user drew an initial set of scribbles to indicate the foreground and background. After using the RFs and CRF to get an initial segmentation, the user gave more scribbles several times for refinement. During each round of refinement, RFs were updated based on the new scribbles and used to predict the probability at each pixel.

The segmentation results were compared with the ground truth for quantitative evaluations. This section uses the Dice similarity coefficient.

$$\text{Dice} = \frac{2|R_s \cap R_g|}{|R_s| + |R_g|} \tag{3.12}$$

where R_s and R_g represent the region segmented by an algorithm and the ground truth, respectively.

Fig. 3.7 shows examples of interactive segmentation of the placenta based on the proposed method. In this segmentation task, scribbles are drawn gradually so that the user can refine an initial segmentation. In Fig. 3.7, the first column shows the initial scribbles and the corresponding probability map given by different RFs. Note that at this start stage, the four compared RFs have similar performances. In the second column, the user gives more scribbles for the background, and the scribbles are highly imbalanced. It can be observed the SF ORF and MP ORF predict more pixels as the background compared with OffBa RF, leading to some under-segmentations. In contrast, the result of DyBa ORF is close to that of OffBa RF, which leads to better segmentation accuracy. Fig. 3.8 shows examples of adult lung segmentation. It can be observed that SP ORF and MP ORF achieve lower performance compared with OffBa RF and DyBa ORF when there is a change of the imbalance ratio of scribbles.

Quantitative evaluations of these two segmentation tasks after the last stage of interaction are listed in Table 3.3 and Table 3.4, respectively. The measurements for evaluation are: G-mean and Dice score (DS) of the probability map thresholded by 0.5, DS after using CRF, and the average update time after the arrival of new scribbles. Table 3.3 and Table 3.4 show that DyBa ORF achieves a higher accuracy than SP ORF and MP ORF, and a comparable accuracy with OffBa RF, with significantly reduced

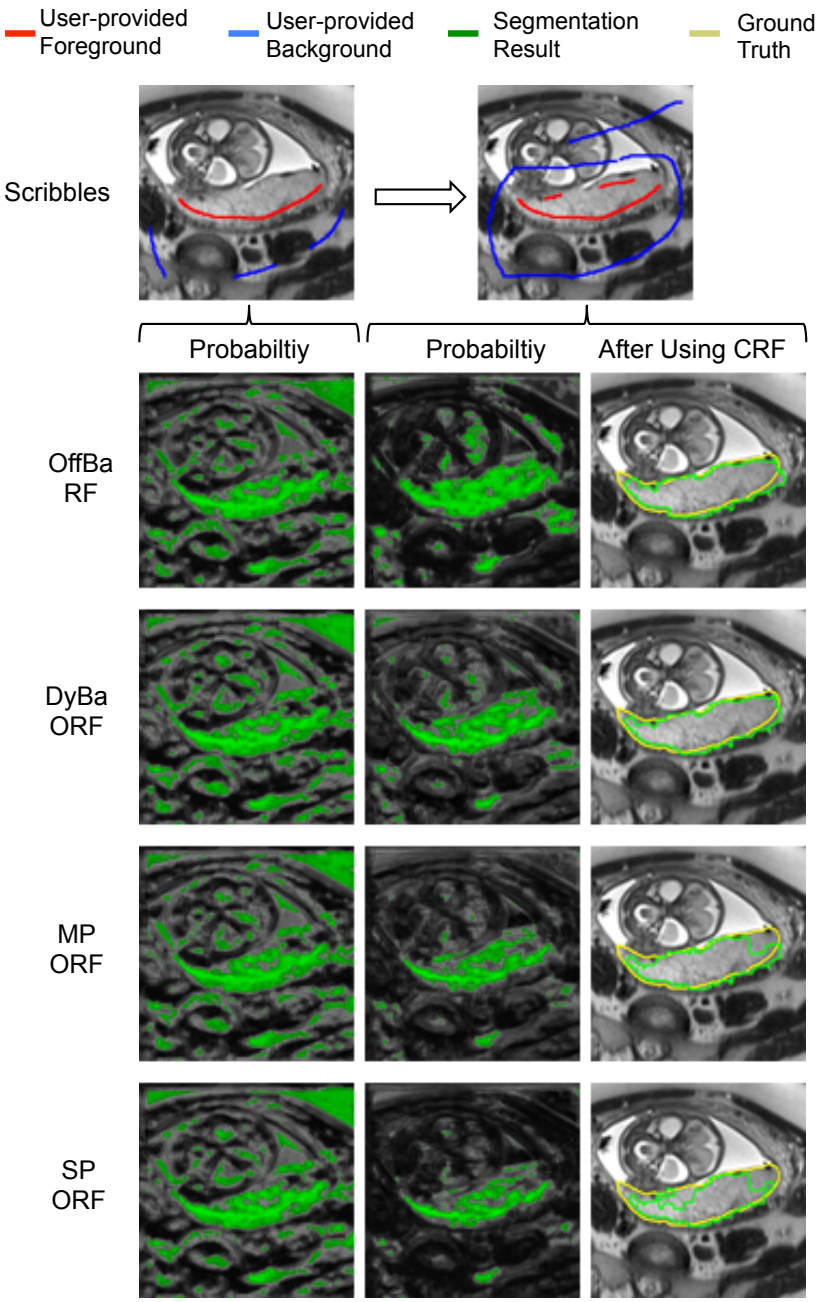


Figure 3.7: Visual comparison of DyBa ORF and counterparts for placenta segmentation from fetal MR images. The first row shows two stages of interaction, where scribbles are extended with a changing imbalance ratio. Probability higher than 0.5 is highlighted by green color. The last column shows the final segmentation and the ground truth.

update time (p-value of Student's t-test<0.05).

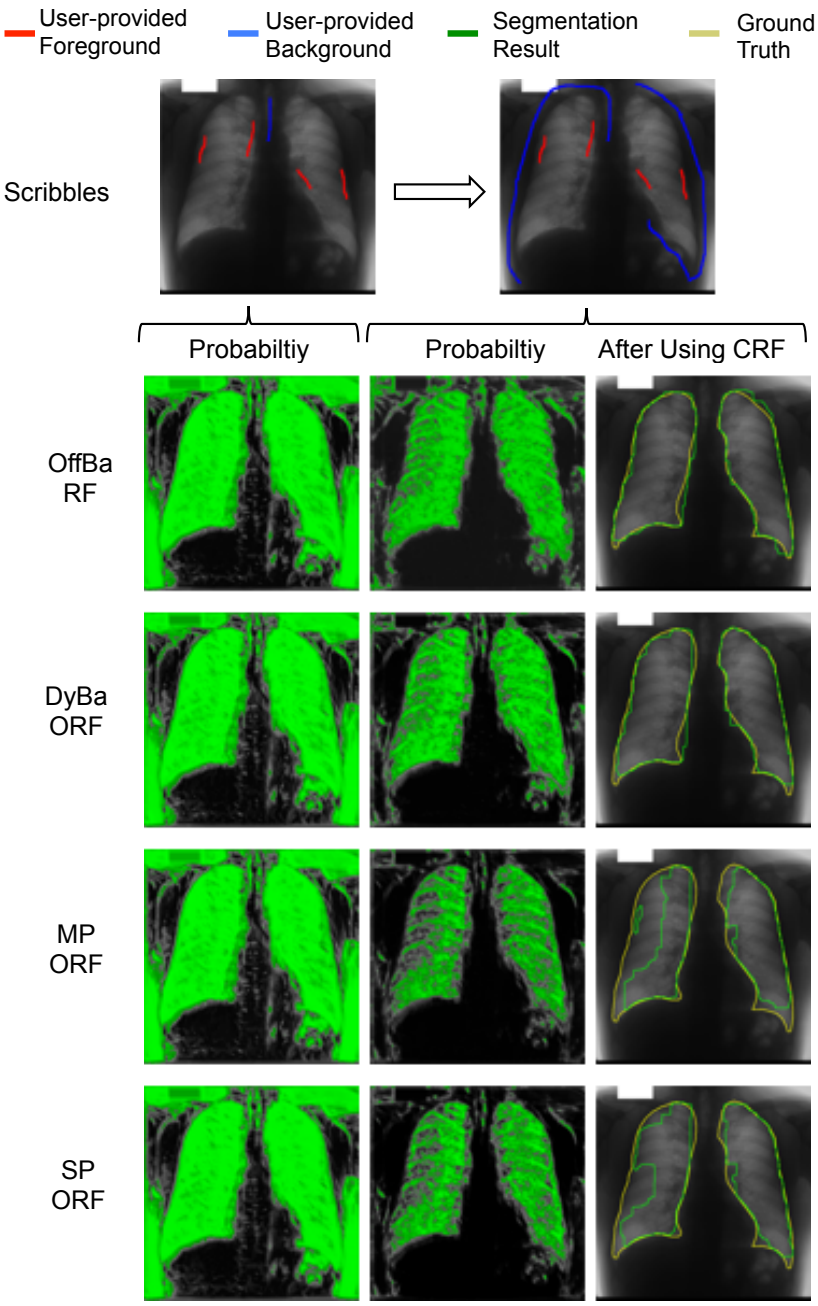


Figure 3.8: Visual comparison of DyBa ORF and counterparts for adult lung segmentation from radiographs. The first row shows two stages of interaction, where scribbles are extended with a changing imbalance ratio. Probability higher than 0.5 is highlighted by green color. The last column shows the final segmentation and the ground truth.

3.4 Discussion and Conclusion

Experimental results show that SP ORF achieves the worst performance, as it does not explicitly deal with data imbalance. MP ORF [153] performs better compared with SP

Table 3.3: G-mean and Dice Score (DS) of DyBa ORF and counterparts for placenta segmentation. G-mean and DS(RF) were measured on probability given by RFs. DS(CRF) was measured on the result after using CRF. t_u is the time for forests update after the arrival of new scribbles. The bold font shows values that are not significantly different from the corresponding results of OffBa RF(p-value of Student's t-test$>$0.05).

Method	G-mean(%)	DS(RF)(%)	DS(CRF)(%)	Average t_u(s)
OffBa RF	84.24±4.02	74.97±7.20	89.32±3.62	1.80±0.92
DyBa ORF	83.09±4.18	**75.25±6.88**	**89.17±3.73**	0.42±0.22
MP ORF	78.21±8.12	71.98±9.76	85.14±9.13	0.37±0.18
SP ORF	74.49±6.94	69.40±8.55	79.32±12.07	0.53±0.26

Table 3.4: G-mean and Dice Score (DS) of DyBa ORF and counterparts for adult lung segmentation. G-mean and DS(RF) were measured on probability given by RFs. DS(CRF) was measured on the result after using CRF. t_u is the time for forests update after the arrival of new scribbles. The bold font shows values that are not significantly different from the corresponding results of OffBa RF(p-value of Student's t-test$>$0.05).

Method	G-mean(%)	DS(RF)(%)	DS(CRF)(%)	Average t_u(s)
OffBa RF	90.80±2.30	86.87±3.89	94.25±1.62	7.40±1.17
DyBa ORF	90.08±2.36	**86.69±3.56**	**94.06±1.64**	1.52±0.43
MP ORF	85.51±3.82	82.95±4.43	90.53±3.59	1.14±0.30
SP ORF	83.38±5.52	80.93±6.70	87.27±9.18	2.19±0.68

ORF, but it fails to be adaptive to imbalance ratio changes. OffBa RF, which learns from scratch for each update, and DyBa ORF, which considers the new imbalance ratio in both existing and new data, are adaptive to imbalance ratio changes. Compared with OffBa RF, DyBa ORF achieves similar accuracy with reduced update time, which shows that DyBa ORF is more suitable for interactive image segmentation. In addition, the results indicate that the SP ORF and MP ORF need some additional user interactions to achieve the same accuracy as obtained by DyBa ORF. This indirectly demonstrates that the proposed model is helpful in reducing user interactions and saving interaction time.

In this chapter, the data imbalance problem is addressed by down-sampling the majority class. In Section 3.2.2.2, the dynamically balanced online Bagging can also be implemented through up-sampling the minority class or weighting different samples. However, up-sampling the minority class leads to more samples and therefore longer time to train and update the trees in the forest compared with down-sampling the majority class. This is less efficient for interactive image segmentation. Weighting

different samples has less randomness compared with the above two alternative methods. In addition, it leads to a weighted version of Information Gain or Gini Index. When the imbalance ratio changes, the weights for different samples and therefore the Information Gain or Gini Index at all the split nodes need to be re-computed in order to be adapted to the new imbalance ratio, which results in a reconstruction of the trees. Thus, this is also not as efficient as down-sampling the majority class.

In conclusion, this chapter presents a Dynamically Balanced Online Random Forest to deal with incremental and imbalanced training data with a changing imbalance ratio, which occurs in the scribble-and-learning-based image segmentation. The proposed method is adaptive to imbalance ratio changes by combining a dynamically balanced online Bagging and a tree growing and shrinking strategy to update the Random Forests. Experimental results show that it achieved a higher accuracy than traditional ORFs, with a higher efficiency than its offline counterpart. Thus, it is more suitable for interactive image segmentation. It can also to be applied to other online learning problems with imbalanced data and a changing imbalance ratio.

Chapter 4

Slic-Seg: Minimally Interactive Segmentation of the Placenta from Sparse and Motion-corrupted Volumetric Images

4.1 Introduction

This chapter relies mostly on materials from my MedIA paper [76], which is extended from a MICCAI paper [75] and a workshop paper [210].

In this chapter, I extend the interactive 2D segmentation method proposed in Chapter 3 to deal with volumetric fetal MR images. I first deal with segmentation from a single volume (3D), and then combine multiple volumes of the same patient (4D) for segmentation.

For single volume segmentation, considering the large inter-slice spacing (i.e., sparse acquisition) and motion between neighboring slices, it is difficult to take advantage of 3D contextual information for segmentation. Therefore, the sparse and motion-corrupted volume is treated as a stack of 2D slices. I propose a slice-by-slice learning-based semi-automatic approach named Slic-Seg that combines high-level features, ORFs and CRFs. It is different from traditional interactive 3D segmentation methods in the following ways: 1) It aims to make better use of user inputs to improve

segmentation accuracy and reduce the number of user interactions. User interactions are only required in a single start slice. The remaining slices in the same volume are segmented sequentially and automatically, without additional user interactions. 2) ORF is employed for efficient learning based on high-level features, allowing the training set to be expanded on the fly, so that the learning can be adapted to the appearance change in different slices. As a result, the method can achieve high performance with a minimal number of user inputs.

The motivation for multi-volume segmentation is that fetal MR images are usually acquired from multiple views that provide complementary resolution. The high intra-slice resolution and low inter-slice resolution make it difficult to get a good segmentation result from a single 3D volume. Thus, combining multiple volumes of the same patient can take advantage of more image contextual information that has a potential to provide better segmentation results. In this chapter, I propose a co-segmentation framework with a probability-based 4D Graph Cuts to utilize the complementary resolution of multiple volumes acquired from different views of the same patient.

4.2 Method

The workflow of the proposed method (Slic-Seg) for single volume segmentation is depicted in Fig. 4.1, and the co-segmentation framework for multiple volumes are shown in Fig. 4.2. For single volume segmentation, the user selects a start slice and draws a few scribbles in that slice to indicate the foreground and the background, respectively. An ORF is used to efficiently learn from these scribbles and predict the probability of each unlabeled pixel belonging to the foreground or the background. That probability is incorporated into a 2D CRF to get the segmentation result of the start slice, based on which new training data are automatically obtained and added to the training set of the ORF on the fly. As shown in Fig. 4.1, to get the segmentation result for a volumetric placenta image, the remaining slices are segmented sequentially and automatically without additional user interactions.

For co-segmentation of multiple volumes of the same patient (Fig. 4.2), each volume is first segmented independently by the single volume Slic-Seg. Then these vol-

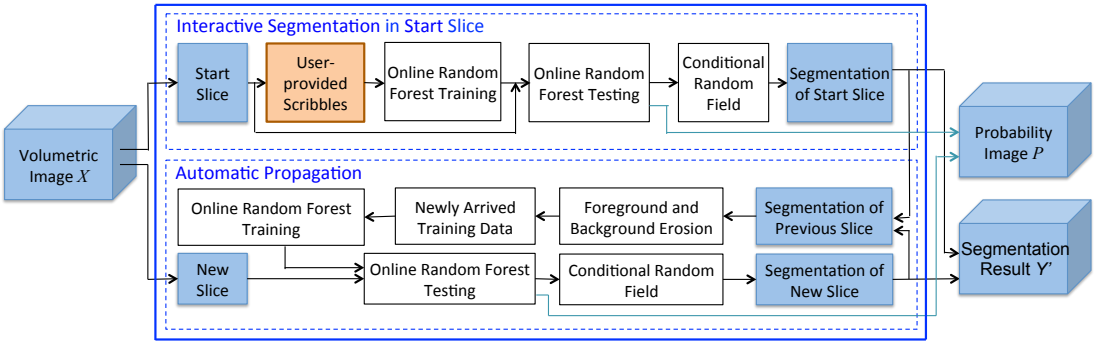


Figure 4.1: The workflow of single volume segmentation by Slic-Seg. User interactions are only required in the start slice. The remaining slices in the same volume are segmented sequentially and automatically with Online Random Forests.

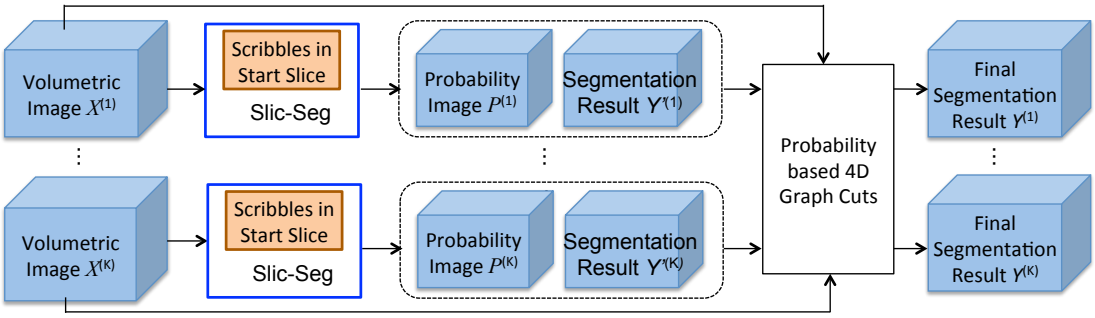


Figure 4.2: Co-segmentation of multiple volumes.

umes are co-segmented by a 4D Graph Cuts framework that takes advantage of the complementary resolution of different volumes and enforces a consistency between them. The 4D Graph Cuts leads to refined segmentation results of all the volumes.

4.2.1 Segmentation of a Single Volume

As a pre-processing step, slices in a stack are rigidly aligned to correct gross motion between them. The 2D rigid alignment is implemented by using ITK[1] [211] with normalized correlation metric and a gradient decent optimizer. Then, histogram matching is implemented to address the different contrast between slices. The features for each pixel are the same as those used in Chapter 3 (Section 3.2.1). The segmentation of a single volume consists of two stages: 1) interactive segmentation of the start slice, and 2) automatic propagation to the remaining slices.

[1] https://itk.org

4.2.1.1 Interactive Segmentation of the Start Slice

The start slice is selected from the middle part of the placenta in the volume, which is easy and convenient for the user to locate. Segmentation of the start slice follows the 2D slice segmentation method presented in the previous chapter (Fig. 3.2). The start slice is segmented through an ORF that learns from scribbles provided by the user. The raw output of the ORF is postprocessed with a 2D CRF for spatial regularization. The user may give additional scribbles to get a good segmentation of the start slice.

4.2.1.2 Automatic Segmentation Propagation

After the segmentation of the start slice, the constructed ORF and the segmentation result of the start slice are used to guide an automatic propagation-based segmentation of the remaining slices. Considering the potential inhomogeneous appearance in different slices, as shown in Fig. 1.7, the ORF learned from the start slice may not work well on a slice far from it. Thus, updating the ORF using training data from more slices can help the ORF be more robust when dealing with slices with different appearances.

During the propagation, after one slice S_i is segmented, new training data are generated based on the segmentation result of S_i. Though the segmentation of S_i may not be very accurate, the central region of the segmented result has a high confidence to be the foreground, and pixels outside the segmentation with a distance to the segmented boundary have a high confidence to be the background. Therefore, these pixels can be used as new training data for the ORF. Morphological erosion operators are employed to get the skeleton of the segmentation result of S_i, and the skeleton is used as new positive training data. The background is also eroded by a morphological operator with a given radius (i.e. 10 pixels), and the erosion result is used as new negative training data.

As shown in Fig. 4.1, the new training data obtained from S_i are added to the existing training set of the ORF on the fly. The ORF is updated and used to test the next slice S_{i+1}. This results in a probability map, which is combined with a 2D CRF to get the label of S_{i+1}. In practice, the segmentation is propagated from the start slice to two ends of the placenta in two directions. A CRF is used in every slice of the volumetric image. The propagation towards either direction stops when the

segmentation result of a new slice does not include the foreground label. After the propagation, the segmentation of all the slices are stacked to construct the volumetric segmentation result.

4.2.1.3 Variants of Single Volume Slic-Seg

In order to analyze how each component of the above described method affects the segmentation, three of its variants are considered for comparison:

Offline Slic-Seg: this variant only leverages user-provides scribbles in the start slice as training data for an offline RF. The offline RF is not updated when the segmentation of a new slice is obtained during the propagation. It uses the same high-level features and CRF as used by the proposed Slic-Seg.

Slic-Seg using low-level features: this variant is the same as the proposed Slic-Seg except that it employs only intensity-based features rather than high dimensional features including GLCM and Haar wavelet.

Slic-Seg without CRF: this method uses the same high-level features and ORF as used by the proposed Slic-Seg, but omits the CRF. To get the binary segmentation label, the output of the ORF is thresholded (threshold probability is 0.5) and then the largest connected component is selected to reduce noises in the prediction of ORF. Then, morphological opening and closing operations based on a square kernel of size 3×3 are used to get a smoothed result.

4.2.2 Co-segmentation of Multiple Volumes

Since the single volume Slic-Seg implements spatial regularization by using the CRF in each 2D slice, the consistency between neighboring slices is not explicitly modeled. In addition, it deals with each volumetric image independently, and the large inter-slice spacing may corrupt segmentation results during the propagation. To address these problems, I propose to refine the segmentation results of single volume Slic-Seg. The refinement step co-segments volumes acquired from different views of the same patient by taking advantage of their complementary resolution in a probability-based 4D Graph Cuts framework, as shown in Fig. 4.2. The framework is general for multiple volumes. In the experiments, this chapter uses the volumes from axial and

sagittal views of the same patient.

4.2.2.1 Preprocessing of Multiple Volumes

Before the co-segmentation, an intra-volume 2D rigid registration is used to align the slices within each volume to alleviate the motion between slices. The rigid registration uses the same method as in Section 4.2.1. Then, an inter-volume 3D registration is used to compensate the motion and deformation between different volumes. The fast Free-Form Deformation (FFD) algorithm [212, 213] is used to register the sagittal view volume to the axial view volume of the same patient. The registration was performed at 3 levels with final grid spacing 6mm$\times$6mm$\times$12mm. The mis-alignment of the placenta between different volumes may not be perfectly addressed due to the complex motion and deformation. Thus, it is more reasonable to not impose the use of a single underlying segmentation (i.e. hard constraint) for all volumes, but rather penalize discrepancies between the segmentation of different volumes after registration (i.e. soft constraint).

4.2.2.2 4D Graph Cuts for Co-segmentation

In Chapter 3, X and Y were used to represent a 2D slice and its label, respectively. In the remaining sections of this chapter, X and Y are used to represent a 3D volume and its 3D labeling result, respectively. Suppose K motion-corrupted volumetric images $X^{(1)}$, $X^{(2)}$, ... $X^{(K)}$ of the same patient that are sparsely acquired from different views (with large inter-slice spacing), the user provides scribbles in a start slice for each volume respectively. These volumes are initially segmented by the single volume Slic-Seg independently. The outputs of Slic-Seg for these volumes are $P^{(1)}$, $Y'^{(1)}$, $P^{(2)}$, $Y'^{(2)}$, ..., $P^{(K)}$, $Y'^{(K)}$ respectively, where $P^{(k)}$ denotes the output probability image for volume k, and $Y'^{(k)}$ is the corresponding segmentation that will be refined in the following step.

To refine these initial segmentation results and get the final labels $Y^{(1)}$, $Y^{(2)}$, ...,

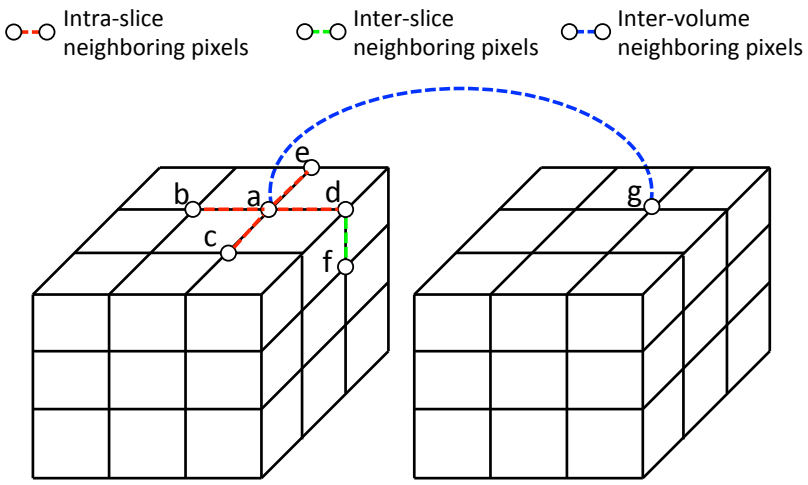


Figure 4.3: Three different kinds of neighboring pixels used in Eq. (4.1). {a, b}, {a, c}, {a, d}, {a, e} are intra-slice neighboring pixels ($\mathcal{N}_1$). {d, f} are inter-slice neighboring pixels ($\mathcal{N}_2$). {a, g} are inter-volume neighboring pixels ($\mathcal{N}_3$).

$Y^{(K)}$, Eq. (3.7) is extended by incorporating inter-slice and inter-volume consistency:

$$E(Y^{(1)},...,Y^{(K)}) = \sum_{k=1}^{K} \sum_{i \in X^{(k)}} \psi(y_i|\mathbf{x}_i, X^{(k)}) + \lambda_1 \sum_{\{i,j\} \in \mathcal{N}_1} B_{i,j} \cdot \delta_{i,j} + \lambda_2 \sum_{\{i,j\} \in \mathcal{N}_2} B'_{i,j} \cdot \delta_{i,j} + \lambda_3 \sum_{\{i,j\} \in \mathcal{N}_3} B''_{i,j} \cdot \delta_{i,j} \quad (4.1)$$

where ψ is defined in a similar way as shown in Eq. (3.8), and it denotes the unary potential based on the prediction of the ORF. $B_{i,j}$, $B'_{i,j}$ and $B''_{i,j}$ are the intra-slice, inter-slice and inter-volume pairwise energy terms, respectively. λ_1, λ_2 and λ_3 are coefficients to adjust the weights of these pairwise energy terms. $\mathcal{N}_1$ is the set of pixel pairs within a 2D slice. $\mathcal{N}_2$ and $\mathcal{N}_3$ are the set of all unordered pairs $\{i, j\}$ of corresponding pixels from two neighboring slices and two volumetric images, respectively.

The three different types of neighboring pixels are depicted in Fig. 4.3, where {a, b}, {a, c}, {a, d} and {a, e} show intra-slice neighboring pixels that belong to $\mathcal{N}_1$. {d, f} shows inter-slice neighboring pixels in a single volume that belong to $\mathcal{N}_2$. {a, g} shows inter-volume neighboring pixels that belong to $\mathcal{N}_3$. To get the inter-image pixel pairs from two volumes $X^{(1)}$ and $X^{(2)}$, for one pixel i in a volume $X^{(k1)}$ ($k1 = 1, 2$), its nearest pixel j in $X^{(1)}$ and $X^{(2)}$ is found, and $\{i, j\}$ is added to $\mathcal{N}_3$ if $j \in X^{(k2)}$ ($k2 = 1, 2$) and $k1 \neq k2$.

$B_{i,j}$ measures the energy due to the intensity difference between two intra-slice neighboring pixels. It is the same function as defined in Eq. (3.10). Here intensity values are used rather than feature values for efficiency. This chapter uses a typical definition of $B_{i,j}$ proposed by Boykov et al. [133].

$$B_{i,j} = \frac{1}{dist(i,j)} \cdot exp\left(-\frac{(x_i - x_j)^2}{2\sigma_1^2}\right) \tag{4.2}$$

where σ_1 controls the sensitivity of difference between x_i and x_j. Due to the inhomogeneous appearance between different slices and between different images, it is less reasonable to define the inter-slice term and inter-image term based on the intensity difference of a pixel pair as in Eq. (4.2). Instead, this chapter uses the difference of foreground probability between a pixel pair to define these two terms. The foreground probability for each pixel is obtained by the ORF prediction in the first phase, i.e., single volume Slic-Seg. The idea is to penalize an inter-slice or inter-image pair of pixels being labeled as the same class when they have a large difference in the probability of being the foreground.

$$B'_{i,j} = \frac{1}{dist(i,j)} \cdot exp\left(-\frac{(p_i^{(k)} - p_j^{(k)})^2}{2\sigma_2^2}\right) \tag{4.3}$$

where $p_i^{(k)} = p(y_i = 1|\mathbf{x}_i, X^{(k)})$, and $\{i,j\} \in \mathcal{N}_2$.

$$B''_{i,j} = exp\left(-\frac{(p_i^{(k1)} - p_j^{(k2)})^2}{2\sigma_3^2}\right) \tag{4.4}$$

where $i \in X^{(k1)},\ j \in X^{(k2)}$, and $\{i,j\} \in \mathcal{N}_3$. σ_2 and σ_3 control the sensitivity of probability difference. The last term in Eq. (4.1) deals with corresponding pixels from different volumes. It is related to the fourth dimension in the 4D Graph Cuts framework and has a different meaning from the first three spatial dimensions. Therefore, the distance between such corresponding pixels is not used to weight the energy in Eq. (4.4). Instead, the weight is set to a constant value and it has been incorporated into λ_3. The energy minimization problem in Eq. (4.1) is solved by the max-flow al-

gorithm [133], after which the final segmentation results of $X^{(1)}$, $X^{(2)}$, ..., $X^{(K)}$ are obtained simultaneously.

4.3 Experiments and Results

4.3.1 Data and Evaluation Methods

Scans of 16 fetuses in the second trimester in two different views were collected using single shot fast spin echo (SSFSE): 1) axial view with slice dimension 512×448, voxel spacing 0.7422mm×0.7422mm, slice thickness 3mm, and 2) sagittal view with slice dimension 256×256, voxel spacing 1.484mm×1.484mm, slice thickness 4mm. The slice number ranges from 50 to 70 among different volumes. For single volume Slic-Seg, a start slice in the middle region of the placenta was selected, and scribbles were provided in the start slice. The algorithm was implemented in C++ with a MATLAB Graphical User Interface (GUI)[2]. Feature extraction was implemented with Compute Unified Device Architecture (CUDA)[3] for a faster speed. The experiments were performed on a Mac laptop (OS X 10.9.5) with 16G RAM and an Intel Core i7 CPU running at 2.5GHz and an NVIDIA GeForce GT 750M GPU. For the ORF, λ was set to 1.0. Tree number was 20 and the maximal tree depth was 10. The minimal sample number for split was 6. The ROI size for feature extraction was 9×9. Parameter setting for 4D Graph Cuts was: K=2, λ_1=40, λ_2=10, λ_3=3, σ_1=2.5, σ_2=0.005, σ_3=0.08. The effect of parameter change on the segmentation performance is presented in Fig. 4.4. It shows that the performance is relatively stable when $\lambda_1 \in [10, 100]$, $\lambda_2 \in [1, 30]$ and $\lambda_3 \in [0.1, 10]$.

Slic-Seg was compared with two other slice-by-slice propagation implementations: an intensity distribution-based Graph Cuts [133] (ID-GC Propagation) and a Geodesic Framework[4] [134] (Geo-Propagation). For ID-GC, the parameter λ mentioned in [133] was set to 10. For Geodesic Framework, there was no parameter tuned by the user. During the propagation, these methods implemented the same morphological operations as in Section 4.2.1.2 on the segmentation of a new slice to generate

[2] Online available: `https://github.com/gift-surg/SlicSeg`
[3] `http://www.nvidia.co.uk/object/cuda-parallel-computing-uk.html`
[4] Implementation from: `http://www.robots.ox.ac.uk/~vgg/software/iseg/`

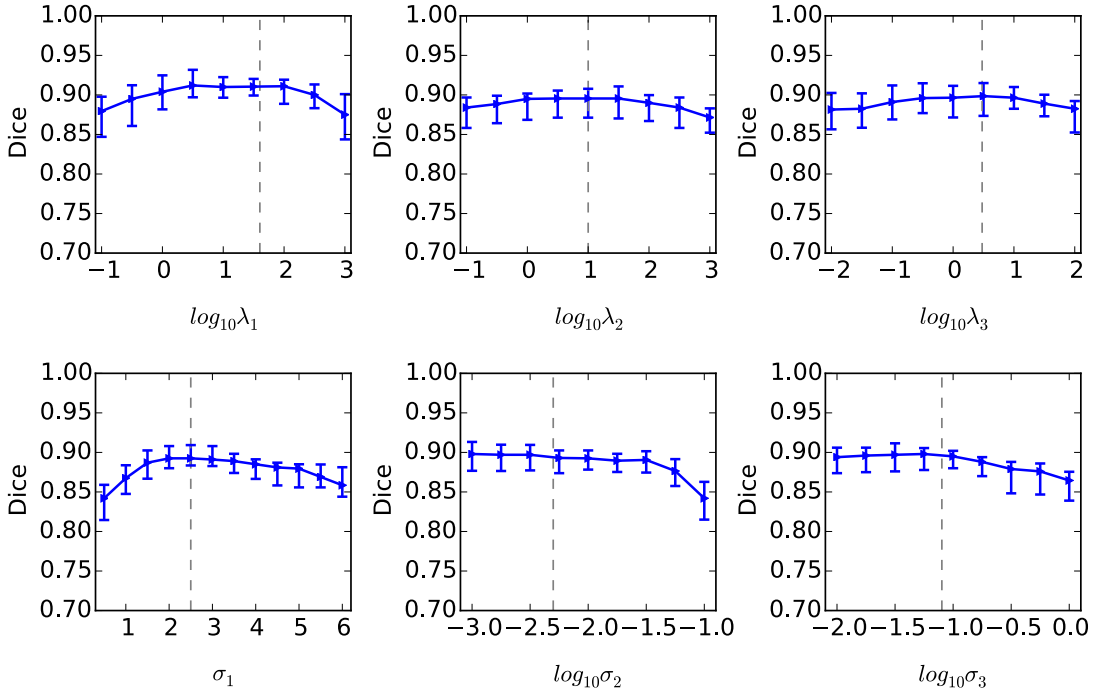


Figure 4.4: The effect of parameter change of Slic-Seg on the segmentation performance. The ranges of λ_1, λ_2, λ_3, σ_2 and σ_3 are denoted by logarithms. The dashed lines indicate the parameter setting in the experiments.

hard constraints for the next slice automatically. Comparisons are also made between Slic-Seg and its three variants: offline Slic-Seg, Slic-Seg using low-level features and Slic-Seg without CRF. All these methods used the same user-provided scribbles in the start slice.

The segmentation results were compared with the ground truth that were manual segmentations given by an experienced Radiologist. For quantitative evaluations, the Dice similarity coefficient defined in Eq. (3.12) and the Average Symmetric Surface Distance (ASSD) were used.

$$ASSD = \frac{1}{|S_s| + |S_g|} \left(\sum_{i \in S_s} d(i, S_g) + \sum_{i \in S_g} d(i, S_s) \right) \tag{4.5}$$

where S_s and S_g represent the set of surface points of the placenta segmented by an algorithm and the ground truth, respectively. $d(i, S_g)$ is the shortest Euclidean distance between the point i and the surface S_g.

To evaluate the intra- and inter-user variability, eight users were asked to perform

the segmentation task independently. Each user provided the scribbles for segmentation twice. The agreement between different segmentation results was measured by Fleiss' kappa coefficient [214]:

$$\kappa = \frac{\bar{P}_a - \bar{P}_e}{1 - \bar{P}_e} \tag{4.6}$$

where $\bar{P}_a$ is the relative observed agreement, and $\bar{P}_e$ is the hypothetical probability of chance agreement. $\bar{P}_a$ and $\bar{P}_e$ are averaged results across all the pixels.

4.3.2 Interactive Segmentation in the Start Slice

Fig. 4.5 shows an example of interactive segmentation in the start slice with scribbles drawn at different positions. It can be observed that with the given scribbles, Slic-Seg achieves the best segmentation accuracy. In addition, Slic-Seg is less sensitive to the position of scribbles than the other methods. Fig. 4.6 shows the effects of different scribble lengths. The first column shows an initial set of scribbles. It can be observed that with the given scribbles, Slic-Seg obtains a result that is close to the ground truth, which outperforms the other alternatives. In the second column, scribbles are extended from those in the first column. The other methods have an improved performance with the extended scribbles, but they still have some mis-segmentations, which require more user interactions to be corrected. This illustrates that Slic-Seg requires fewer scribbles to get good segmentation in the start slice than the other methods.

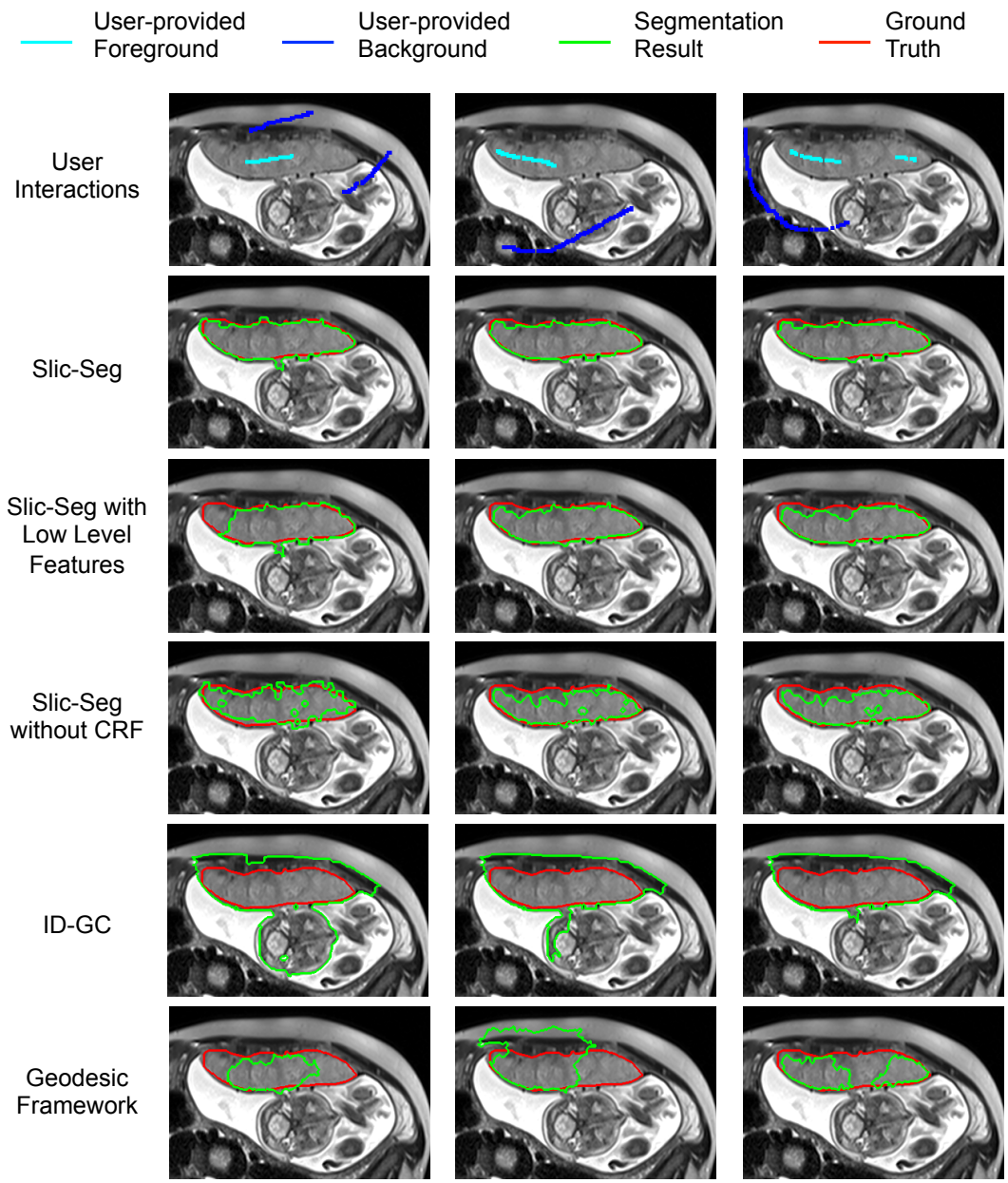


Figure 4.5: Segmentation of the placenta by different methods in the start slice with scribbles drawn at different positions. Note the better segmentation of Slic-Seg compared with the other methods.

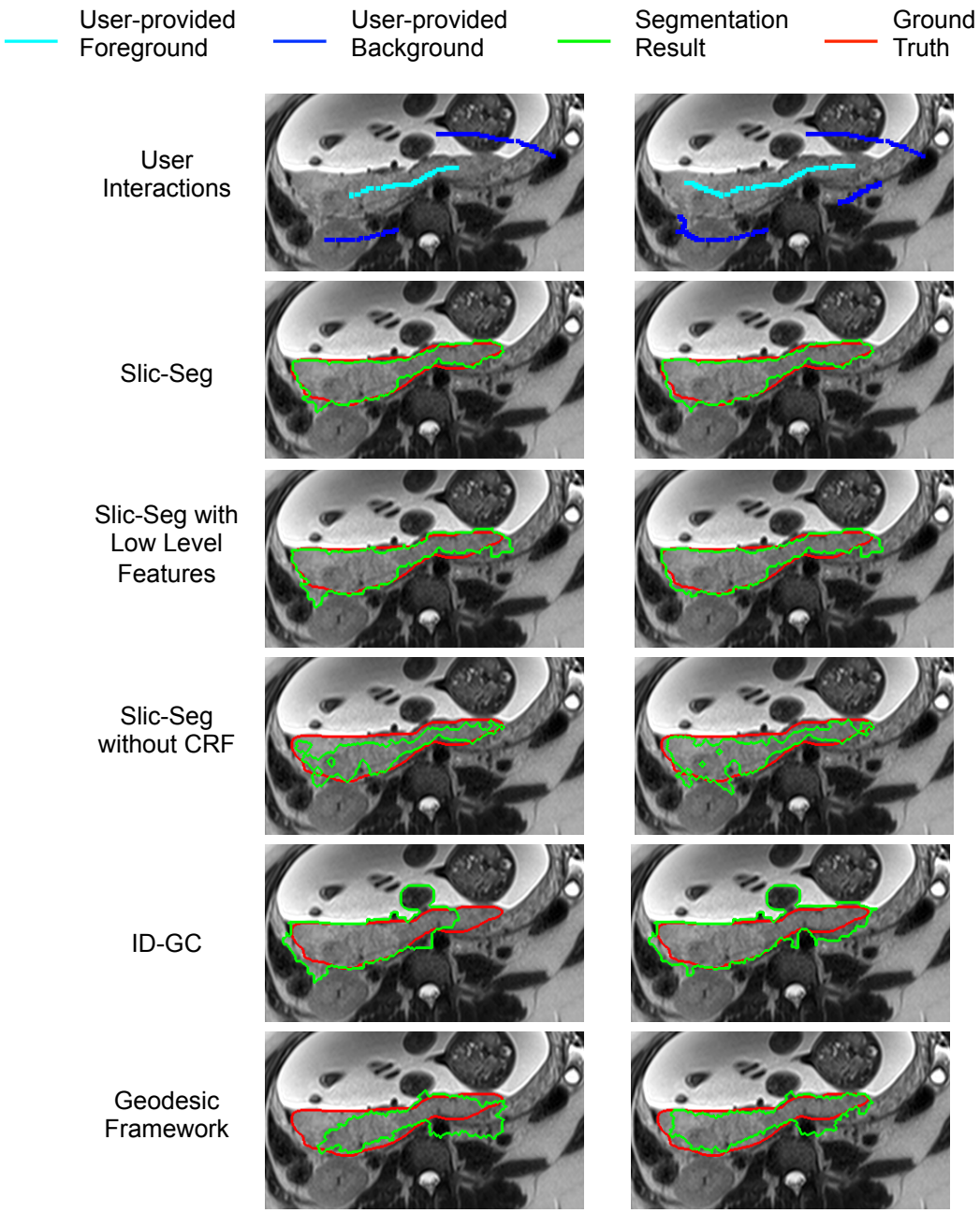


Figure 4.6: Segmentation of the placenta by different methods in the start slice with different scribble lengths.

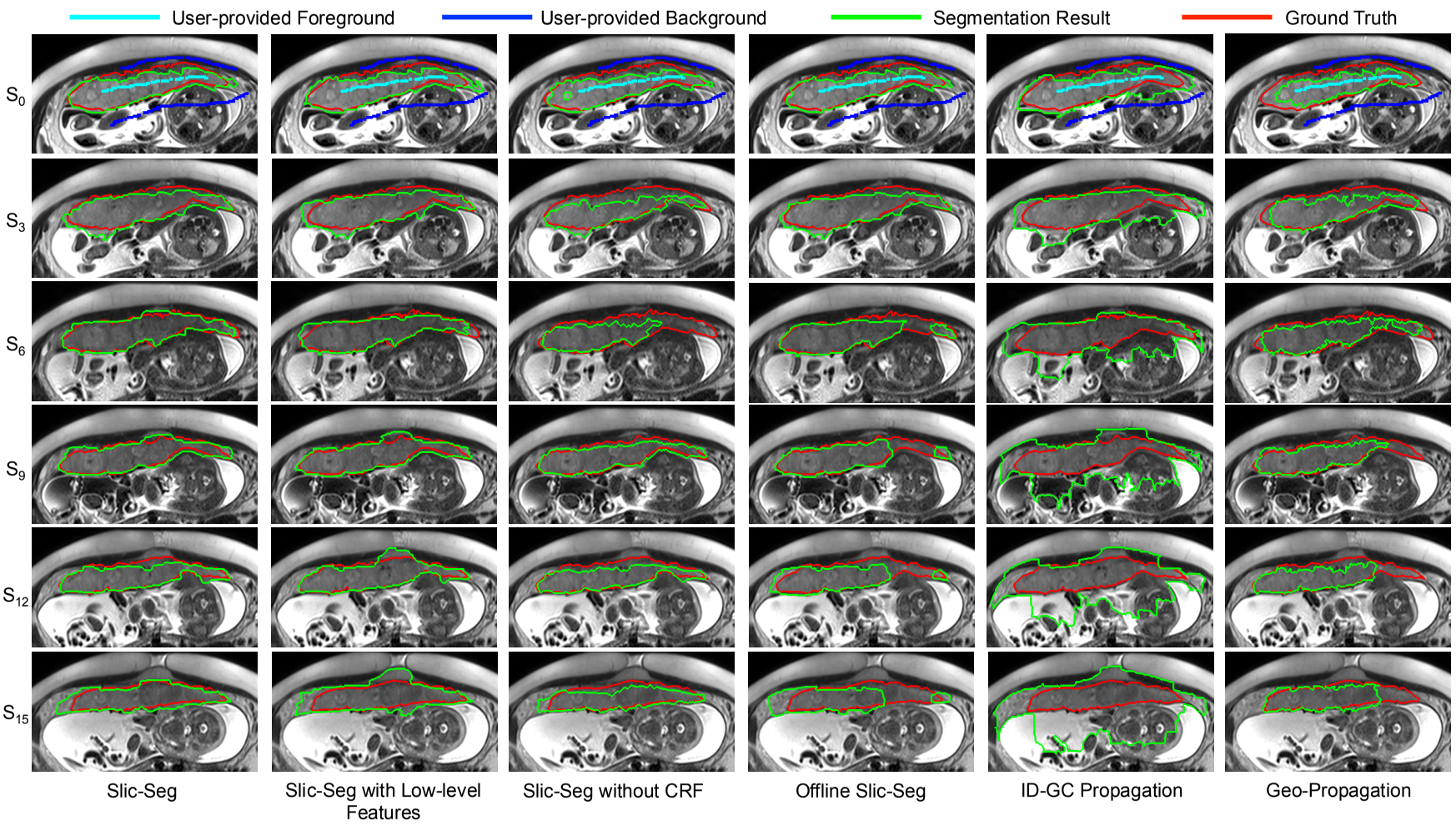


Figure 4.7: Segmentation propagation in a volume by different methods with the same start slice and scribbles. S_i represents the ith slice following the start slice S_0. Scribbles in S_0 are extensive and all the compared methods have a good performance in S_0. During the propagation, only Slic-Seg keeps a high performance.

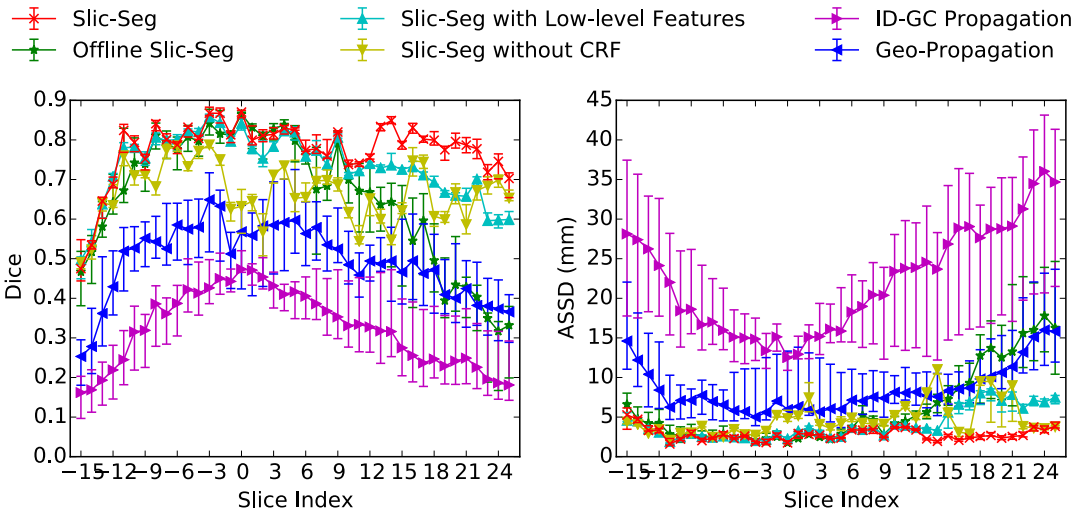


Figure 4.8: Quantitative evaluation of slice-by-slice propagation for placenta segmentation from one volume. The scribbles were only provided in the start slice that is denoted by slice index 0. The evaluation was based on the segmentation results given by eight users.

4.3.3 Automatic Propagation in the Remaining Slices

Fig. 4.7 shows an example of the propagation of different methods with the same user inputs (scribble length: 495 mm) in the start slice S_0. S_i represents the ith slice following the start slice. In Fig. 4.7, though a good segmentation is obtained in the start slice due to an extensive set of scribbles, the errors of offline Slic-Seg, Geo-Propagation and ID-GC Propagation become increasingly large during the propagation. For Slic-Seg with low-level features, in a slice that is close to the start slice (e.g. $i \leq 6$), it can obtain good results. When a new slice is further away from the start slice (e.g. $i \geq 12$), it fails to track the placenta with high accuracy. For Slic-Seg without CRF, the performance fluctuates during the propagation. In contrast, Slic-Seg has a more stable and higher performance during the propagation.

Fig. 4.8 shows the Dice coefficient and ASSD for each slice in one volumetric image which was segmented by all the eight users. For each slice, error bars are used to show the first quartile, median and the third quartile of the Dice coefficient and ASSD. Fig. 4.8 shows that Slic-Seg and its variants have a better performance in the start slice and during the propagation than Geo-Propagation and ID-GC Propagation. Offline Slic-Seg and Slic-Seg with low-level features have a decreased accuracy in

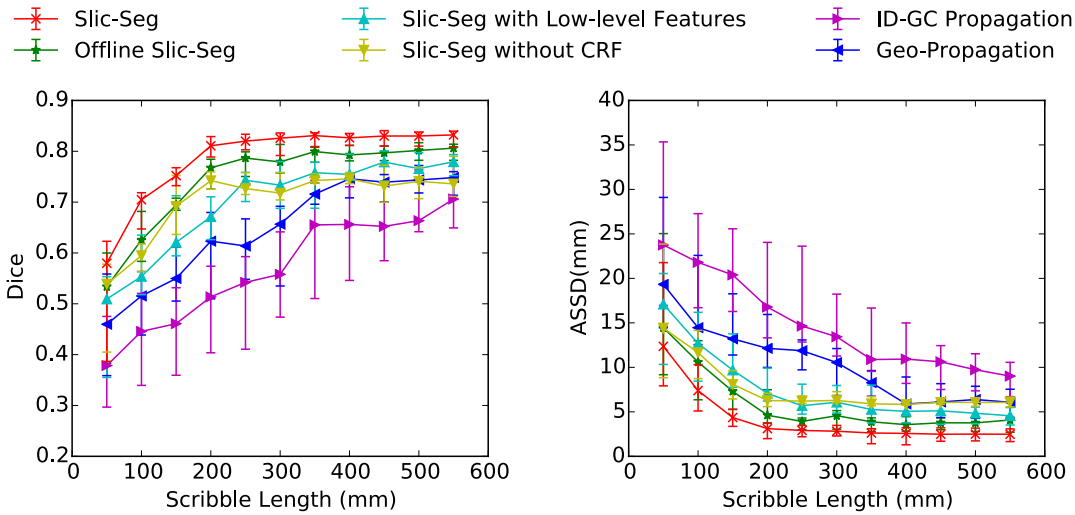


Figure 4.9: The change of Dice (left) and ASSD (right) with increasing length of scribbles that were provided in the start slice. The performance was evaluated for the segmentation of one volume with scribbles given by eight users.

remote slices. The fluctuating performance of Slic-Seg without CRF is also obvious in Fig. 4.8. The comparison shows that Slic-Seg outperforms the other methods during the propagation. In addition, the lower dispersion of Slic-Seg indicates its reduced variability between users compared with the other counterparts.

4.3.4 Interactivity and User Variability

Fig. 4.9 shows the effects of scribble length on the accuracy for segmentation of the total volume. During the user's drawing scribbles, the order of points on the scribbles for the foreground and the background was recorded, and these recorded scribbles were used sequentially and incrementally for segmentation, with the length changing from 50mm to 550 mm. It can be seen in Fig. 4.9 that Slic-Seg achieves a higher accuracy than the others, with its Dice and ASSD plateauing when the length of scribbles was extended to around 200-300mm. Fig. 4.9 also shows the use of ORF, high-level features and CRF improves the segmentation accuracy.

Since the number of slices containing the placenta varies among different volumes, the runtime of propagation-based segmentation is measured in terms of the average runtime for propagation per slice, which is defined as the ratio of the total propagation time for the volume to the number of slices containing the placenta in

Table 4.1: Average runtime per slice (in seconds) for the propagation using different methods. The feature extractions for Slic-Seg and its variants are GPU-based, and the propagations of all the methods are CPU-based.

Slic-Seg	Offline Slic-Seg	Slic-Seg with low-level features	Slic-Seg without CRF	ID-GC Propagation	Geo-Propagation
1.05±0.13	0.84±0.06	0.55±0.10	0.93±0.08	0.12±0.04	0.61±0.07

Table 4.2: Intra- and inter-operator variability of Slic-Seg for segmentation of volumetric images. κ is the Fleiss's kappa coefficient defined in Eq. 4.6.

User	Dice	ASSD(mm)	κ
1	0.81±0.02	2.73±0.62	0.931
2	0.82±0.03	2.57±0.60	0.936
3	0.81±0.03	2.75±0.61	0.949
4	0.80±0.03	2.81±0.73	0.941
5	0.82±0.02	2.58±0.61	0.948
6	0.82±0.02	2.63±0.61	0.945
7	0.82±0.02	2.61±0.74	0.941
8	0.81±0.03	2.76±0.67	0.936
All	0.82±0.02	2.67±0.63	0.932

that volume. The time consumption by the compared algorithms is listed in Table 4.1. Note that the feature extractions for Slic-Seg and its variants are implemented on a GPU, and the propagations of all the methods are implemented on a CPU. Table 4.1 shows ID-GC Propagation has the shortest runtime, and Slic-Seg has a longer runtime that is 1.05±0.13s per slice but still acceptable for interactive segmentation.

The mean value and standard deviation of Dice and ASSD, as well as the intra- and inter-user Fleiss' kappa coefficient are presented in Table 4.2, which shows a low intra- and inter-user variability of Slic-Seg. The quantitative measurement across all the users is 0.82 ± 0.02 in terms of Dice, and 2.67 ± 0.63mm in terms of ASSD. In addition, the intra-user κ ranges from 0.931 to 0.949, and the inter-user κ is 0.932, which indicates the proposed interactive segmentation method has high intra- and inter-user agreement with low variability.

4.3.5 Co-segmentation of Volumes in Multiple Views

After the two volumetric images acquired in axial and sagittal views of the same patient are segmented by single volume Slic-Seg respectively, the initial segmentation results are refined by a co-segmentation step with the proposed 4D probability-based (4D PR)

Table 4.3: Quantitative comparison of different interactive segmentation methods for single volume segmentation. The axial view images have a high axial-view resolution and a low sagittal-view resolution. The sagittal view images have a low axial-view resolution and a high sagittal-view resolution. The best value in each row is shown in bold font.

		ITK-SNAP	GeoS	3D ID-GC	Grow Cut	Slic-Seg
Axial	Dice	0.79±0.03	0.81±0.03	0.79±0.02	0.80± 0.03	**0.82±0.02**
	ASSD (mm)	2.94±0.72	2.68±0.67	3.19±0.61	2.78± 0.66	**2.35±0.47**
	Time (m)	1.98±0.25	2.78±0.82	3.13±0.50	2.84± 0.39	**1.36±0.29**
Sagittal	Dice	**0.81±0.02**	0.79±0.03	0.79±0.02	0.78± 0.03	0.81±0.03
	ASSD (mm)	**2.73±0.48**	3.40±0.76	3.57±0.96	2.99± 0.85	2.84±0.54
	Time (m)	1.78±0.27	1.70±0.65	1.63±0.18	2.01± 0.19	**0.80±0.23**

Graph Cuts. The proposed refinement method is compared with three variants: 3D probability-based refinement (3D PR) using Graph Cuts, 3D intensity-based refinement (3D IR) and 4D intensity-based refinement (4D IR) using Graph Cuts. The 3D methods only consider a single volume for refinement, and the intensity-based methods define the inter-slice and inter-image binary terms based on pixel intensity rather than the probability obtained by the ORF.

Fig. 4.10 shows an example of the initial segmentation by Slic-Seg and the refined results by 3D/4D IR/PR respectively. Image $X^{(1)}$ and $X^{(2)}$ are acquired in axial and sagittal views of the same patient, respectively. $X^{(1)}$ has a high resolution in axial view with a low resolution in sagittal view. $X^{(2)}$ has a low resolution in axial view with a high resolution in sagittal view. The first row shows the initial segmentations of $X^{(1)}$ and $X^{(2)}$. They have some errors compared with the ground truth. The following rows show the refined segmentation results by the four different refinement methods. The dark orange arrows in each row indicate differences between the initial segmentation and the refined results. For the intensity-based methods, although some errors in the initial segmentation are corrected (the dark orange arrows in the last column), additional mis-segmentations are introduced (highlighted by the cyan arrows). Thus, these two methods fail to improve segmentation accuracy. In contrast, the probability-based methods improve the segmentation without causing extra errors. The last two rows show 4D PR outperforms 3D PR in the refinement stage.

For quantitative evaluations, the proposed Slic-Seg with slice-propagation is compared with four other popular interactive methods for single volume segmentation:

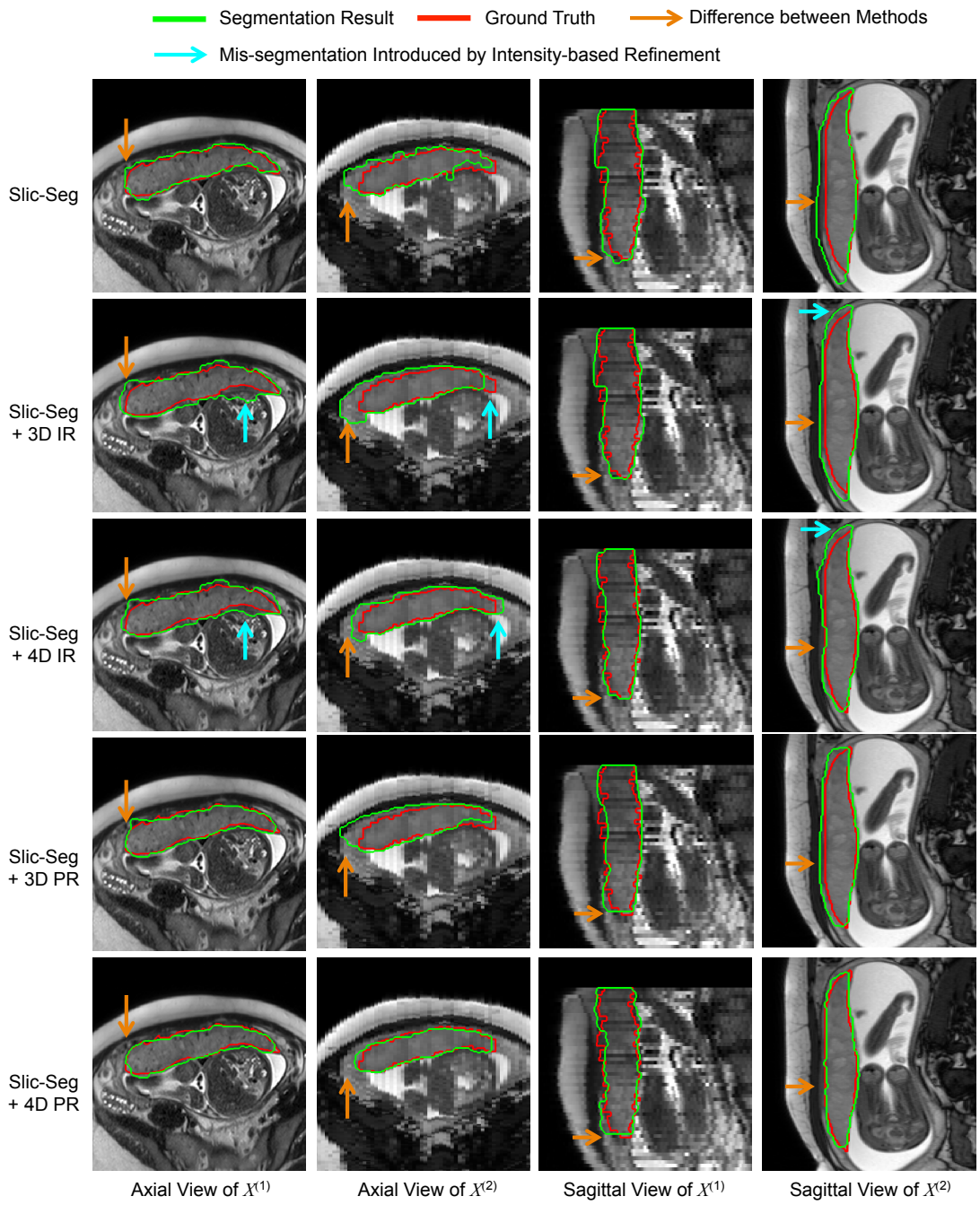


Figure 4.10: Visual comparison of initial segmentation by single volume Slic-Seg and refinement by 3D/4D Graph Cuts using intensity/probability respectively. $X^{(1)}$ and $X^{(2)}$ are acquired in two views of the same patient with complementary resolution. IR and PR refer to intensity- and probability-based refinements, respectively.

ITK-SNAP [138], GeoS [140], 3D ID-GC [133] and GrowCut [215]. For these four methods that are not designed to accept scribbles only in a start slice, scribbles are provided in multiple slices, and after the initial segmentation the user can provide more

Table 4.4: Quantitative comparison of refinement methods based on co-segmentation. The axial view images have a high axial-view resolution and a low sagittal-view resolution. The sagittal view images have a low axial-view resolution and a high sagittal-view resolution. The best value in each row is shown in bold font.

		Slic-Seg	3D IR	4D IR	3D PR	4D PR
Axial	Dice	0.82±0.02	0.80±0.03	0.81±0.02	0.87± 0.03	**0.89±0.02**
	ASSD (mm)	2.35±0.47	3.28±0.62	3.00±0.46	2.16± 0.26	**1.89±0.39**
	Time (m)	**1.36±0.29**	1.83±0.36	2.03±0.40	1.79± 0.38	1.96±0.43
Sagittal	Dice	0.81±0.03	0.80±0.04	0.81±0.03	0.86± 0.02	**0.88±0.02**
	ASSD (mm)	2.84±0.54	3.29±0.72	2.95±0.58	2.41± 0.45	**1.99±0.38**
	Time (m)	**0.80±0.23**	1.08±0.25	1.47±0.28	1.14± 0.26	1.40±0.30

scribbles and execute the algorithm again to correct the result. The results after several rounds of correction when the user accepts are used for evaluation.

Quantitative evaluations of these different interactive methods are shown in Table 4.3, and a comparison between variants of the co-segmentation method is shown in Table 4.4. Table 4.3 and Table 4.4 list the evaluation results of images acquired in both axial and sagittal views. Table 4.3 shows that Slic-Seg requires less time for segmentation than the other methods. For images acquired in axial view, Slic-Seg achieves the best accuracy. For images acquired in sagittal view, Slic-Seg is better than the others except ITK-SNAP. However, Slic-Seg is not significantly lower than ITK-SNAP (p-value of Student's t-test>0.05), and it only requires scribbles in a start slice while ITK-SNAP needs user interactions in multiple slices.

The refined results of Slic-Seg by co-segmentation using 4D PR (Table 4.4) are better than that of ITK-SNAP (Tabel 4.3). Table 4.4 shows Slic-Seg with 4D PR has a better performance than the other counterparts of co-segmentation. In terms of the refinement, 3D IR and 4D IR achieves lower Dice values and higher ASSD values compared with the initial segmentation obtained by single volume Slic-Seg, which indicates that they fail to improve the segmentation accuracy. In contrast, accuracy of the probability-based refinement methods is higher than that of single volume Slic-Seg, and 4D PR performs better than 3D PR. The p-value between them is 6.9×10^{-11} in terms of Dice and 1.1×10^{-10} in terms of ASSD.

4.4 Discussion and Conclusion

For the interactive segmentation with propagation for single volume segmentation, the experiments show that Slic-Seg achieves higher accuracy than Geodesic Framework and Graph Cuts when scribbles are given only in a start slice. The latter two methods rely on low-level gradient or intensity information to model the placenta and the background, which may not be accurate enough in fetal MR images with poor 3D quality. Slic-Seg uses high-level features of multiple aspects including intensity, texture and wavelet coefficients. This provides a better description of differences between the placenta and the background, which outperforms Slic-Seg with low-level features. In addition, online training of the ORF overcomes the potential appearance change when the slice-by-slice segmentation propagates to a remote slice, and the employment of CRF leads to a spatially regularized result of the ORF prediction. These factors allow Slic-Seg to have a good performance during the propagation. Although the use of high-level features increases the computational time, the average runtime of Slic-Seg for one slice is 1.05s, which is acceptable for interactive segmentation. In addition, it is possible to pre-compute the features so that runtime can be reduced during the propagation.

The experiments show that with the increase of scribble length, better segmentation results are achieved by all the compared methods. However, Slic-Seg requires fewer user interactions to reach the plateau accuracy. This results in the minimization of user interactions, considering it only needs user-provided scribbles in the start slice. Besides, Table 4.2 shows high intra- and inter-operator agreements, which indicates a low variability within and between users.

There are three reasons to refine the segmentation results of single volume Slic-Seg for placenta segmentation from fetal MR images. First, the large inter-slice spacing and inhomogeneous appearance between slices make accurate segmentation hard to achieve from a single volumetric image. Second, single volume Slic-Seg applies CRFs only in 2D slices, without taking into account the inter-slice connectivity, which may lead to jagged surfaces in 3D space. In addition, post-segmentation refinement can be helpful considering errors in the automatic propagation. The skeleton of the foreground

and eroded background in a segmented slice are used to guide the segmentation of the following slice, which makes the error in a slice less likely to be propagated to its following slice. As is shown in Fig. 4.8, the propagation of Slic-Seg is robust in most slices, and the accumulated error becomes large only in terminal slices due to a large change of the shape of the placenta between two sequential slices. It has been shown that the proposed automatic refinement leveraging multiple volumes with 4D Graph Cuts can reduce errors related to the initial propagation.

The proposed refinement method combines the complementary resolution of images acquired in different views, and reduces the segmentation errors by incorporating inter-slice and inter-image consistency. The experiment shows intensity-based 3D and 4D Graph Cuts do not improve the segmentation accuracy, indicating that sole intensity information is not sufficient for good segmentation. In contrast, by defining the inter-slice and inter-image pairwise energy based on probability obtained by the ORF using high-level features, a large improvement of accuracy is obtained as shown in Table 4.4. In addition, the 4D PR achieves a higher improvement in the refinement step than 3D PR, which demonstrates the co-segmentation of two images leads to higher accuracy than using a single volumetric image. In the current co-segmentation implementation, the $\mathcal{N}_3$ neighborhoods are defined based on the nearest voxels from different volumes. Considering the potential alignment error, the method might be improved by defining the inter-image neighborhoods based on the voxels in a local area weighted by the distance or similarity, therefore mutual information or patch-based analysis [216] might be helpful for a more robust segmentation. Note that although two images are co-segmented in the experiment, the proposed method is formulated in Eq. (4.1) so that it can deal with more volumetric images.

Though the ORF-based methods presented in Chapter 3 and Chapter 4 are efficient for online learning from a single image, they have two limitations to deal with a large number of images. First, the features used by ORFs are manually designed for a local patch. These features are selected based on experience but may not be the most effective ones. Second, these methods either learn from a single slice or a single volume, while ignoring the information from other patients. With the availability of

data from a large number of patients, learning from a set of data can help the classifier become more robust and capable to generalize when dealing with images of different patients. Such limitations can be addressed by Deep Learning [217, 218], which can learn the most suitable features automatically from a large set of annotated training data.

In conclusion, this chapter presents an interactive, learning-based method for placenta segmentation from sparse and motion-corrupted fetal MR volumes. The proposed method can deal with segmentation from a single volume and multiple volumes, respectively. A slice-by-slice propagation method using ORFs and CRFs is used to segment a single volume. It only requires user inputs in a start slice, and the remaining slices are segmented sequentially and automatically to get a volumetric segmentation. The segmentation can be further refined by co-segmentation of multiple volumes in different views of the same patient using a probability-based 4D Graph Cuts method. Experimental results demonstrate the proposed segmentation framework has a stable performance between and within users, and the co-segmentation achieves a large improvement of accuracy compared with single-volume segmentation. Therefore, this approach might be suitable for segmentation of the placenta in planning systems for fetal and maternal surgery, and for rapid characterization of the placenta by MR images. Its first clinical application might be fetoscopic placement optimization in the treatment of twin-twin transfusion syndrome.

Chapter 5

Deep Interactive Geodesic Framework for Placenta Segmentation

5.1 Introduction

The work presented in this chapter is from my article published in TPAMI [77].

This chapter revisits the problem of 2D segmentation of the placenta studied in Chapter 3. Rather than learning from a single slice, this chapter uses CNNs to learn from a set of annotated images to obtain accurate and robust segmentation results with reduced user interactions.

This chapter aims to integrate user interactions into a CNN framework to obtain accurate and robust segmentation of medical images and, at the same time, this chapter aims to make the interactive framework more efficient with a minimal number of user interactions by using CNNs. With the good performance CNNs have shown in automatic image segmentation tasks [122, 123, 125, 121, 128], they have the potential to reduce the number of user interactions for interactive image segmentation. However, only a few works have been reported on applying CNNs to interactive segmentation tasks [129, 154, 155, 156].

The contributions of this chapter are four-fold. 1). A deep CNN-based interactive framework is proposed for medical image segmentation. It uses one CNN to get an initial automatic segmentation, which is refined by another CNN that takes as input the initial segmentation and user interactions; 2). This chapter presents a new way to com-

bine user interactions with CNNs based on geodesic distance maps that are used as extra channels of the input for CNNs. The experimental results show that using geodesic distance can lead to improved segmentation accuracy compared with using Euclidean distance; 3). A resolution-preserving CNN structure is proposed and it leads to more accurate segmentation results compared with traditional CNNs with resolution loss, and 4). The current Recurrent Neural Network (RNN)-based CRFs [171] for segmentation is extended so that the back-propagatable CRFs can employ user interactions as hard constraints and all the parameters of potential functions can be trained in an end-to-end way. Experimental results show the new method achieves a large improvement from automatic CNNs, and obtains comparable accuracy for placenta segmentation with fewer user interventions and less user time compared with traditional interactive methods. Appendix A demonstrates that the proposed method also works well on clavicle segmentation from radiographs. In Appendix B, it is shown that this method can be extended to a 3D version with validations on 3D brain tumor segmentation from adult MR images.

5.2 Method

The proposed deep interactive segmentation method based on CNNs and geodesic distance transforms (DeepIGeoS) is depicted in Fig. 5.1. To minimize the number of user interactions, DeepIGeoS uses two CNNs: an initial segmentation proposal network (P-Net) and a refinement network (R-Net). P-Net takes as input a raw image with m channels and gives an initial automatic segmentation. Then the user checks the segmentation and provides some interactions (clicks or scribbles) to indicate mis-segmented regions. R-Net takes as input the original image, the initial segmentation and the user interactions to provide a refined segmentation. P-Net and R-Net use a resolution-preserving structure that captures high-level features from a large receptive field without the loss of resolution. They share the same structure except the input of R-Net has $m+3$ channels. Based on the initial automatic segmentation obtained by P-Net, the user might give clicks/scribbles to refine the result more than one time through R-Net. Differently from previous works [74] that re-train the learning model

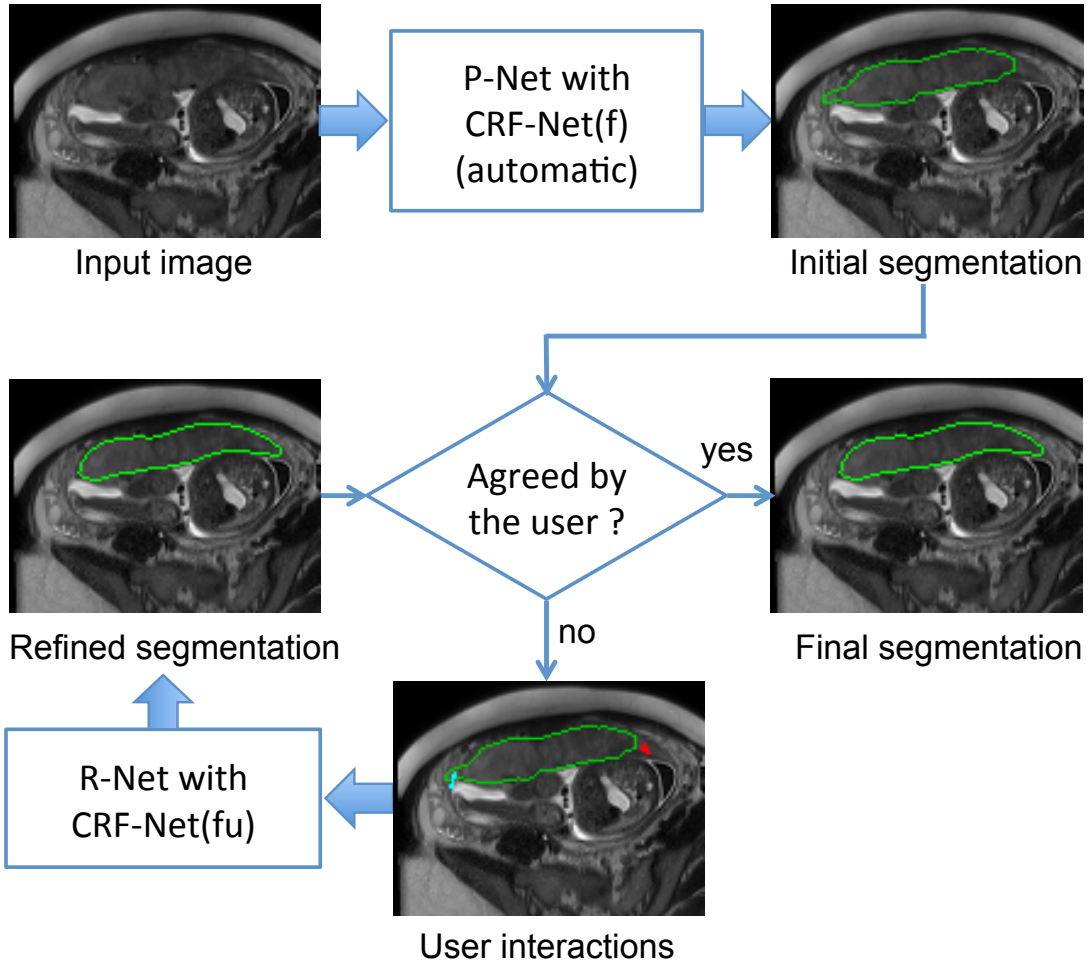


Figure 5.1: Overview of the proposed deep interactive segmentation method (DeepIGeoS). P-Net automatically proposes an initial segmentation that is refined by R-Net with user-interactions indicating mis-segmentations. CRF-Net(f) is the proposed back-propagatable CRF that uses freeform pairwise potentials. It is extended to CRF-Net(fu) that uses user interactions as hard constraints.

each time when new user interactions are given, the proposed R-Net is only trained with user interactions once, since it takes a considerable time to re-train a CNN model with a large training set.

To make the segmentation result more spatially consistent and to use scribbles as hard constraints, both P-Net and R-Net are connected with a CRF, which is modeled as an RNN (CRF-Net) so that it can be trained jointly with P-Net/R-Net by back-propagation. Freeform pairwise potentials are used in the CRF-Net. The way user interactions are used is presented in 5.2.1, and the structures of P-Net and R-Net are detailed in 5.2.2. In 5.2.3, the implementation of the proposed CRF-Net is described.

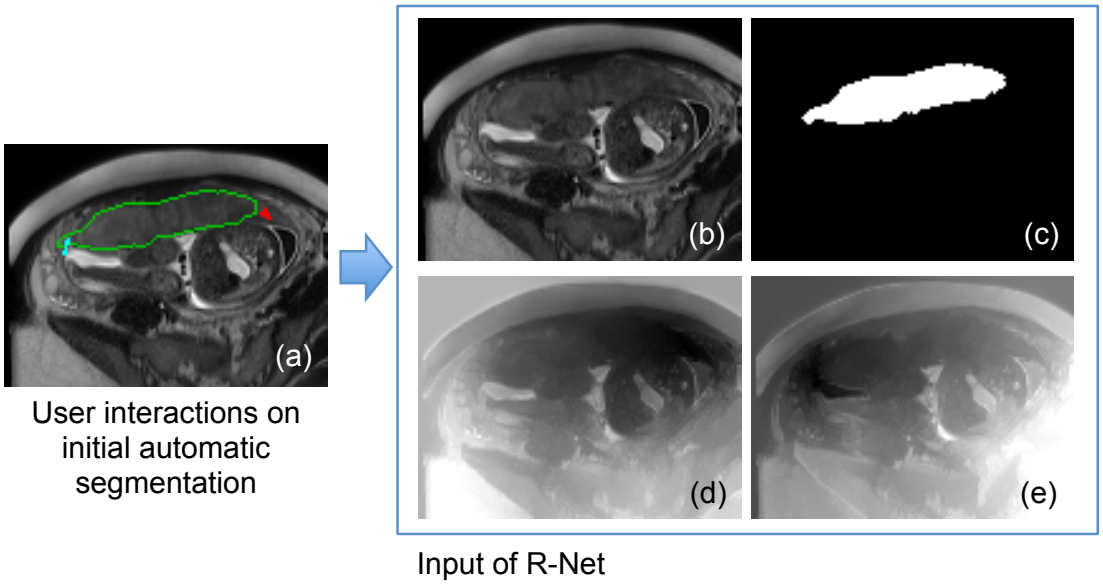


Figure 5.2: Input of R-Net using geodesic distance transforms of user interactions. (a) The user gives clicks/scribbles to correct foreground(red) and background(cyan) on the initial segmentation. (d) and (e) are geodesic distance maps based on foreground and background interactions, respectively. The original image (b) is combined with the initial automatic segmentation (c) and geodesic distance maps (d), (e) by channel-concatenation, and the concatenated output is used as the input of R-Net.

5.2.1 User Interactions-based Geodesic Distance Maps

In the proposed method, scribbles are provided by the user to refine an initial automatic segmentation obtained by P-Net. A scribble labels a set of pixels as the foreground or background. Interactions with the same label are converted into a distance map. In [156], the Euclidean distance was used due to its simplicity. However, the Euclidean distance treats each direction equally and does not take the image context into account. In contrast, the geodesic distance helps to better differentiate neighboring pixels with different appearances, and improves label consistency in homogeneous regions [140]. GeoF [115] uses the geodesic distance to encode variable dependencies in the feature space and it is combined with Random Forests for semantic segmentation. However, it is not designed to deal with user interactions. This chapter proposes to encode user interactions via geodesic distance transforms for CNN-based segmentation.

Suppose S_f and S_b represent the set of pixels belonging to foreground scribbles and background scribbles, respectively. Let i be a pixel in an image X, then the un-

signed geodesic distance from i to the scribble set $S(S \in \{S_f, S_b\})$ is:

$$G(i,S,X) = \min_{j \in S} D_{geo}(i,j,X) \tag{5.1}$$

$$D_{geo}(i,j,X) = \min_{p \in \mathscr{P}_{i,j}} \int_0^1 \|\nabla X(p(s)) \cdot \mathbf{u}(s)\| ds \tag{5.2}$$

where $\mathscr{P}_{i,j}$ is the set of all paths between pixel i and j. p is one feasible path and it is parameterized by $s \in [0,1]$. $\mathbf{u}(s)$ is a unit vector that is tangent to the direction of the path and is defined as $\mathbf{u}(s) = p'(s)/\|p'(s)\|$. If no scribbles are drawn for either the foreground or the background, the corresponding geodesic distance map is filled with random numbers.

Fig. 5.2 shows an example of the geodesic distance transforms of user interactions. The geodesic distance maps of user interactions and the initial automatic segmentation have the same height and width as X. They are concatenated with the raw channels of X so that a concatenated image with $m+3$ channels is obtained, which is used as the input of the refinement network R-Net.

5.2.2 P-Net: Resolution Preserving 2D CNN using Dilated Convolution

CNNs in the proposed method are designed to capture high-level features from a large receptive field without the loss of resolution of the feature maps. They are adapted from VGG-16 [119] and made resolution-preserving. Fig. 5.3 shows the structure of P-Net. It consists of six blocks of layers. The first and second blocks have two convolution layers respectively, and each of the following three blocks has three convolution layers. The convolution kernels in the first five blocks have a fixed size 3×3 and a fixed number of output channels $C = 64$. The stride of each convolution layer is set to 1. The five blocks have dilation parameters of 1, 2, 4, 8 and 16, respectively, so they capture features at different scales. Features from these five blocks are concatenated and fed into the sixth block that serves as a classifier. In the sixth block, two dropout layers are used to prevent over-fitting, and two convolution layers are used to

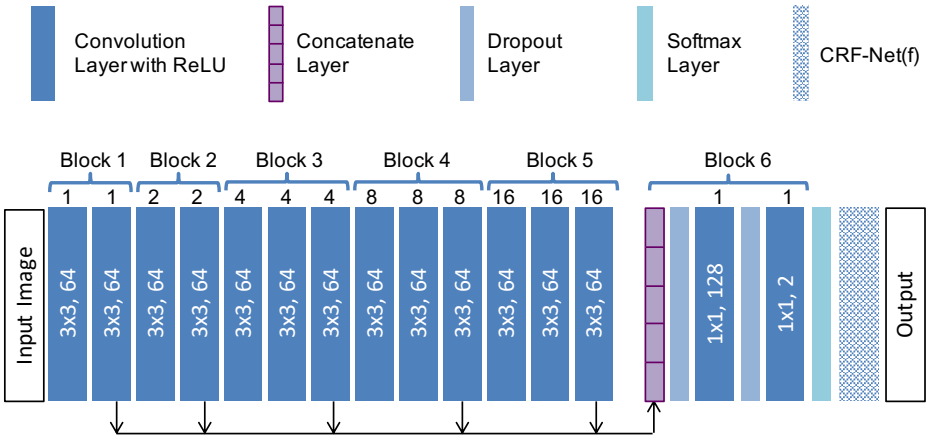


Figure 5.3: Structure of P-Net with CRF-Net(f). The numbers in each dark blue box denote convolution kernel size and number of output channels. The number on the top denotes dilation parameter. The stride of each convolution layer is set to 1 so that the resolution is kept the same through the network. The R-Net uses the same structure except its input has three additional channels shown in Fig. 5.2 and the CRF-Net(f) is replaced by CRF-Net(fu) (Section 5.2.3). In Chapter 6, P-Net is also used for bounding box-based 2D segmentation and extended for 3D segmentation.

map the concatenated features to a classification score for each pixel corresponding to the foreground or the background class. These two convolution layers use convolution kernels of size 1×1 and dilation parameter 1, and their output channels are 128 and 2 respectively. A softmax layer is used after the sixth block to convert the classification scores to probabilities of belonging to different classes.

In order to get a more spatially consistent segmentation and add hard constraints when scribbles are given, a CRF is applied on the basis of the output from block 6. The CRF is implemented by a recurrent neural network (CRF-Net, detailed in 5.2.3), which can be jointly trained with R-Net. The CRF-Net gives a regularized prediction for each pixel, which is fed into a cross entropy loss function layer during training.

R-Net uses the same structure as P-Net except that its number of input channels is m+3 and it employs user interactions in the CRF-Net. To obtain an exponential increase of the receptive field, VGG-16 uses a max-pooling and downsampling layer after each block. However, this implementation would decrease the resolution of feature maps exponentially. Therefore, to preserve resolution through the network, the proposed method removes the max-pooling and downsampling layers and uses dilated convolution in each block.

Let X be a 2D image of size $W \times H$, and let K_{rq} be a square dilated convolution kernel with a size of $(2r+1)\times(2r+1)$ and a dilation parameter of q, where $r \in \mathbb{Z}$ and $q \in \mathbb{Z}$. The dilated convolution of X with K_{rq} is defined as:

$$X_c(x,y) = \sum_{i=-r}^{r} \sum_{j=-r}^{r} X(x-qi,\ y-qj) K_{rq}(i+r,\ j+r) \tag{5.3}$$

For the proposed P-Net/R-Net, r is set to 1 for block 1 to block 5, so the size of a convolution kernel becomes 3×3. The dilation parameter in block i is set to:

$$q_i = d \times 2^{i-1},\ i = 1,\ 2,\ ...,\ 5 \tag{5.4}$$

where $d \in \mathbb{Z}$ is a system parameter controlling the base dilation parameter of the network. d is set to 1 in the experiments.

The receptive field of a convolution kernel K_{rq} is $(2rq+1)\times(2rq+1)$. Let $R_i \times R_i$ denote the receptive field of block i, and R_i can be computed as:

$$R_i = 2\Big(\sum_{j=1}^{i} \tau_j \times (rq_j)\Big) + 1,\ i = 1,\ 2,\ ...,\ 5 \tag{5.5}$$

where τ_j denotes the number of convolution layers in block j, with a value of 2, 2, 3, 3, 3 for the five blocks respectively. When $r = 1$, the receptive field size of each block is $R_1 = 4d+1$, $R_2 = 12d+1$, $R_3 = 36d+1$, $R_4 = 84d+1$, $R_5 = 180d+1$, respectively. Thus, these blocks can capture features at different scales.

5.2.3 CRF-Net: Back-propagatable CRF with Freeform Pairwise Potential and User Constraints

In [171], a CRF based on RNN was proposed and it can be trained by back-propagation. Rather than using Gaussian functions, this chapter extends this CRF so that the pairwise potentials can be freeform functions, which is referred to as CRF-Net(f). In addition, this chapter integrates user interactions in the CRF-Net(f) in the interactive refinement context, which is referred to as CRF-Net(fu). The CRF-Net(f) is connected to P-Net and the CRF-Net(fu) is connected to R-Net.

Let Y be the label map assigned to an image X and the label set be $\mathscr{L}$ = {0, 1, ..., $L-1$ }. The Gibbs distribution $P(Y=y|X) = \frac{1}{Z(X)}\exp(-E(y|X))$ models the probability of Y given X in a CRF, where $Z(X)$ is the normalization factor known as the partition function, and $E(y)$ is the Gibbs energy:

$$E(y) = \sum_i \psi(y_i) + \sum_{(i,j)\in\mathscr{N}} \phi(y_i, y_j) \tag{5.6}$$

where the unary potential $\psi(y_i)$ measures the cost of assigning label y_i to pixel i, and the pairwise potential $\phi(y_i, y_j)$ is the cost of assigning labels y_i, y_j to pixel pair i, j. $\mathscr{N}$ is the set of all pixel pairs. In the proposed method, the unary potential is obtained from the P-Net or R-Net that gives initial scores of different classes for each pixel. The pairwise potential is defined as:

$$\phi(y_i, y_j) = \mu(y_i, y_j) f(\tilde{\mathbf{f}}_{ij}, d_{ij}) \tag{5.7}$$

where d_{ij} is the Euclidean distance between pixels i and j. $\mu(y_i, y_j)$ is the compatibility between the label of i and that of j, and represented by a matrix of size $L \times L$. $\tilde{\mathbf{f}}_{ij} = \mathbf{f}_i - \mathbf{f}_j$, where $\mathbf{f}_i$ and $\mathbf{f}_j$ represent the feature vectors of i and j, respectively. The feature vectors can either be learned by a network or be derived from image features such as spatial location with intensity values. For experiments the latter one is used in this chapter, as in [171, 127, 133] for simplicity and efficiency. $f(\cdot)$ is a function in terms of $\tilde{\mathbf{f}}_{ij}$ and d_{ij}. Instead of defining $f(\cdot)$ as a single Gaussian function [133] or a combination of several Gaussian functions [171, 127], this chapter defines it as a freeform function represented by a fully connected neural network (Pairwise-Net) that can be learned during training. The structure of Pairwise-Net is shown in Fig. 5.4. The input is a vector composed of $\tilde{\mathbf{f}}_{ij}$ and d_{ij}. There are two hidden layers and one output layer.

Graph Cuts [133, 162] can be used to minimize Eq. (5.6) when $\phi(\cdot)$ is submodular such as when the segmentation is binary with $\mu(\cdot)$ being the delta function and $f(\cdot)$ being positive [161]. However, this is not the case for the proposed method since it learns $\mu(\cdot)$ and $f(\cdot)$ where $\mu(\cdot)$ may not be the delta function and $f(\cdot)$ could be negative. Continuous max-flow [159] can also be used for the minimization, but its parameters

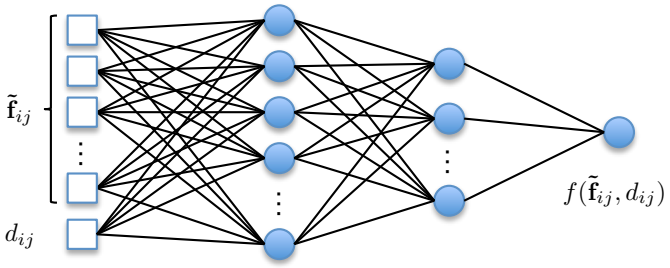


Figure 5.4: Structure of Pairwise-Net for pairwise potential function $f(\tilde{\mathbf{f}}_{ij}, d_{ij})$. $\tilde{\mathbf{f}}_{ij}$ is the difference of features between a pixel pair i and j. d_{ij} is the Euclidean distance between them.

are manually designed. Alternatively, mean-field approximation [171, 127, 172] is often used for efficient inference of the CRF while allowing learning parameters by back-propagation. Instead of computing $P(Y|X)$ directly, an approximate distribution $Q(Y|X) = \prod_i Q_i(y_i|X)$ is computed so that the KL-divergence $\mathbf{D}(Q||P)$ is minimized. This yields an iterative update of $Q_i(y_i|X)$ [171, 127, 172].

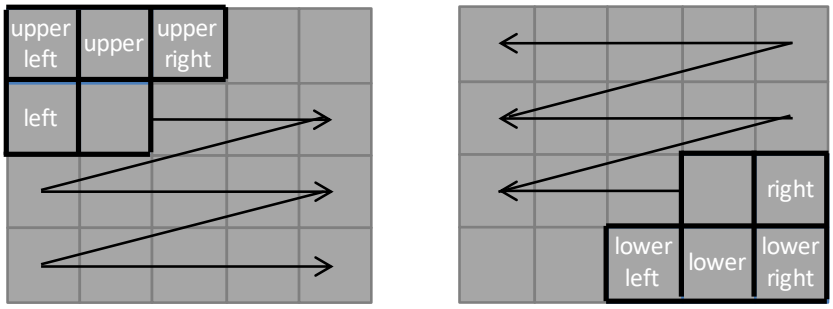
$$Q_i(y_i|X) = \frac{1}{Z_i} e^{-E(y_i)} = \frac{1}{Z_i} e^{-\psi(y_i) - \phi(y_i)} \tag{5.8}$$

$$\phi(y_i = l|X) = \sum_{l' \in \mathscr{L}} \mu(l, l') \sum_{j \in \mathscr{N}_i} f(\tilde{\mathbf{f}}_{ij}, d_{ij}) Q_j(l'|X) \tag{5.9}$$

where $\mathscr{L}$ is the label set, and $\mathscr{N}_i$ is the set of neighboring pixels of i. For the proposed CRF-Net(fu), with the set of user-provided scribbles $S_{fb} = S_f \cup S_b$, the probability of pixels in the scribble set is forced to be 1 or 0. The following equation is used as the update rule for each iteration:

$$Q_i(y_i|X) = \begin{cases} 1 & \text{if } i \in S_{fb} \text{ and } y_i = s_i \\ 0 & \text{if } i \in S_{fb} \text{ and } y_i \neq s_i \\ \frac{1}{Z_i} e^{-E(y_i)} & \text{otherwise} \end{cases} \tag{5.10}$$

where s_i denotes the user-provided label of a pixel i that is in the scribble set S_{fb}. Q is updated through a multi-stage mean-field method in an RNN following the implemen-



(a) forward pass (b) backward pass

Figure 5.5: Fast geodesic distance transforms based on raster-scan. At each position of forward pass (a) or backward pass (b), the gradients between current pixel and its four neighborhoods within the kernel are calculated.

tation in [171]. Each mean-field layer splits Eq. 5.8 into four steps including message passing, compatibility transform, adding unary potentials and normalizing [171].

5.2.4 Implementation Details

The computation of geodesic distance transforms follows the raster-scan algorithm proposed in [140] that is fast due to accessing the image memory in contiguous blocks. As shown in Fig. 5.5, this method calculates the geodesic distance by applying a forward pass scanning and a backward pass scanning with a 3×3 kernel. In the forward pass, the image is scanned from the top-left to the bottom-right corner. The upper-left, upper, upper-right and left components of the image gradient ∇X are computed. In the backward pass, the image is scanned from the bottom-right to top-left corner. Image gradients in terms of right, lower-left, lower, lower-right are computed. More precise geodesic distance can be obtained by using larger kernels. As suggested by [140], a 3×3 kernel is used for its efficiency and good performance.

For the proposed CRF-Net with freeform pairwise potentials, two observations motivate the use of pixel connections based on local patches instead of full connections within the entire image. First, the permutohedral lattice implementation [127, 171] allows efficient computation of fully connected CRFs only when pairwise potentials are Gaussian functions. However, a method that relaxes the pairwise potentials as freeform functions represented by a network (Fig. 5.4) cannot use that implementation

and therefore would be inefficient for fully connected CRFs. Suppose an image with a size of $M \times N$, a fully connected CRF has $MN(MN-1)$ pixel pairs. For a small image with $M = N = 100$, the number of pixel pairs would be almost 10^8, which requires not only large amount of memory but also long computational time. Second, though long-distance dependency helps to improve segmentation in most RGB images [127, 171, 125], this would be very challenging for medical images since the contrast between the target and background is often low [176]. In such cases, long-distance dependency may lead the label of a target pixel to be corrupted by the large number of background pixels with similar appearances. Therefore, to maintain a good efficiency and avoid long-distance corruptions, the pairwise connections for one pixel are defined within a local patch centered on that. In the experiment, the patch size is set to 7×7 based on experience.

$\mu(\cdot)$ is initialized as $\mu(y_i, y_j) = [y_i \neq y_j]$, where $[\cdot]$ is the Iverson Bracket [171]. A fully connected neural network (Pairwise-Net) with two hidden layers is used to learn the freeform pairwise potential function (Fig. 5.4). The first and second hidden layers have 32 and 16 neurons, respectively. In practice, this network is implemented by an equivalent fully convolutional neural network with 1×1 kernels. A pre-training step is used to initialize the Pairwise-Net with an approximation of a contrast sensitive function [133]:

$$f_0(\tilde{\mathbf{f}}_{ij}, d_{ij}) = \exp\left(-\frac{||\tilde{\mathbf{f}}_{ij}||^2}{2\sigma^2 \cdot F}\right) \cdot \frac{\omega}{d_{ij}} \tag{5.11}$$

where F is the dimension of feature vector $\mathbf{f}_i$ and $\mathbf{f}_j$, and ω and σ are two parameters controlling the magnitude and shape of the initial pairwise function, respectively. In this initialization step, σ is set to 0.08 and ω is set to 0.5 based on experience. Similar to [127, 125, 171], $\mathbf{f}_i$ and $\mathbf{f}_j$ are set as values in input channels (i.e, image intensity in this case) of P-Net for simplicity of implementation and for obtaining contrast-sensitive pairwise potentials. To pre-train Pairwise-Net, a training set $T' = \{X', Y'\}$ of size 100k is generated, where X' is the set of features simulating the concatenated $\tilde{\mathbf{f}}_{ij}$ and d_{ij}, and Y' is the set of prediction values simulating $f_0(\tilde{\mathbf{f}}_{ij}, d_{ij})$. For each sample

s in T', the feature vector x'_s has a dimension of $F+1$ where the first F dimensions represent the value of $\tilde{\mathbf{f}}_{ij}$ and the last dimension denotes d_{ij}. The c-th channel of x'_s is filled with a random number k', where $k' \sim Norm(0,2)$ for $c \leq F$ and $k' \sim U(0,8)$ for $c = F+1$. The ground truth of prediction value y'_s for x'_s is obtained by Eq. (5.11). After generating X' and Y', a Stochastic Gradient Descent (SGD) algorithm with a quadratic loss function is used to pre-train the Pairwise-Net so that it is initialized with an approximation of Eq. (5.11).

For pre-processing, all the images are normalized by the mean value and standard deviation of training set. Training images are augmented by vertical or horizontal flipping, random rotation with angle range $[-\pi/8, \pi/8]$, random zoom with scaling factor range [0.8, 1.25]. The cross entropy loss function and SGD algorithm are used for optimization, with minibatch size 1, momentum 0.99 , weight decay 5×10^{-4}. The learning rate is halved every 5k iterations. Since a proper initialization of the P-Net and CRF-Net(f) is helpful for a faster convergence of the joint training, the P-Net with CRF-Net(f) is trained in three steps. First, the P-Net is pre-trained with initial learning rate 10^{-3} and maximal number of iterations 100k. Second, the Pairwise-Net in the CRF-Net(f) is pre-trained as described above. Third, the P-Net and CRF-Net(f) are jointly trained with initial learning rate 10^{-6} and maximal number of iterations 50k.

After the training of P-Net with CRF-Net(f), simulation of user interactions is implemented for the training of R-Net with CRF-Net(fu). First, P-Net with CRF-Net(f) is used to obtain an automatic segmentation for each training image. The segmentation is compared with the ground truth to find mis-segmented regions. Then the user interactions on each mis-segmented region are simulated by randomly sampling n pixels in that region. Suppose the size of one connected under-segmented or over-segmented region is N_m, n for that region is set to 0 if $N_m < 30$ and $\lceil N_m/100 \rceil$ otherwise based on experience. Examples of simulated user interactions on a training image are shown in Fig. 5.6. With these simulated user interactions on the initial segmentation of training data, the training of R-Net with CRF-Net(fu) is implemented through SGD, which is similar to the training of P-Net with CRF-Net(f).

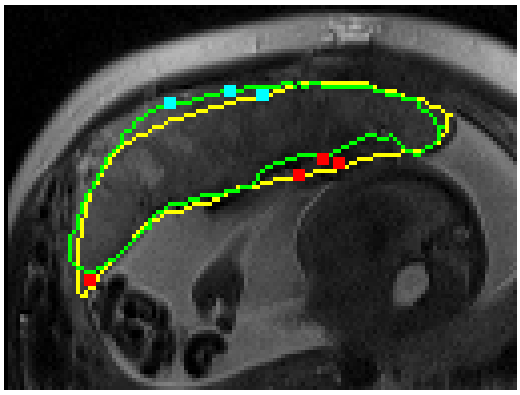


Figure 5.6: Simulated user interactions on a training slice. Green: automatic segmentation given by P-Net with CRF-Net(f). Yellow: ground truth. Red and cyan dots are simulated clicks on under-segmentations and over-segmentations, respectively.

The Caffe[1] [219] deep learning library was used to implement the proposed P-Net and R-Net with CRF-Net. The training process was done on the UK Emerald cluster[2] via a single node with two 8-core E5-2623v3 Intel Haswells and two K80 NVIDIA GPUs and 128GB memory. The testing process with user-interactions was performed on a Mac laptop (OS X 10.9.5) with 16G RAM and an Intel Core i7 CPU running at 2.5GHz and an NVIDIA GeForce GT 750M GPU. A Matlab GUI was developed for the interactive segmentation task.

5.3 Experiments

5.3.1 Data and Comparison Methods

For the experiments, MRI scans for 25 pregnant women in the second trimester were collected with SSFSE. The data were acquired in axial view with pixel size between 0.74 mm$\times$0.74 mm and 1.58 mm$\times$1.58 mm and slice thickness 3 - 4 mm. Each slice was resampled with a uniform pixel size of 1 mm$\times$1 mm and cropped by a box of size 172$\times$128 containing the placenta. 17 volumes with 624 slices were used for training. Three volumes with 122 slices were used for validation and five volumes with 179 slices were used for testing. The ground truth was manually delineated by an experienced Radiologist.

The performance of the proposed P-Net was compared with that of FCN [123] and

[1] `http://caffe.berkeleyvision.org`
[2] `http://www.ses.ac.uk/high-performance-computing/emerald`

DeepLab [125]. Pre-trained models of FCN[3] and DeepLab[4] based on ImageNet[5] [220] were fine-tuned for placenta segmentation from fetal MR images. These two networks were extended from VGG-16 [119] so that they allow obtaining the label of an image through one single forward pass. However, the output resolution of these two networks was 1/8 of the input resolution, thus the outputs were zoomed in by a factor of 8 to obtain a segmentation with the original resolution. For fine-tuning, the "step" SGD optimization method was used with initial learning rate 10^{-5}, maximal iterations 100k. The step policy was the same as that used for P-Net training. Since the input of FCN and DeepLab should have three channels, each of the gray-level images was duplicated twice and concatenated into a three-channel image as the input. The P-Net was also compared with its variant P-Net(b5) that only uses features from block5 (Fig. 5.3) instead of concatenated multi-scale features.

The proposed CRF-Net(f) with freeform pairwise potentials was compared with two counterparts: 1). Dense CRF as an independent post-processing step for the output of P-Net. The implementation presented in [127, 125] was used in experiments. Instead of being learned by back-propagation, the parameters of this CRF were manually tuned based on a coarse-to-fine search scheme as suggested by [125]. 2). CRF-Net(g) which refers to the CRF that can be trained jointly with CNNs by using Gaussian pairwise potentials [171].

Three methods of dealing with user interactions were compared. 1). Min-cut user-editing [136], where the initial probability map (output of P-Net in this case) is combined with user interactions to solve an energy minimization problem with min-cut [133]; 2). Using the Euclidean distance of user interactions in R-Net, which is referred to as R-Net(Euc), and 3). The proposed R-Net with the geodesic distance of user interactions.

DeepIGeoS was also compared with four other interactive segmentation methods: 1). Geodesic Framework [221] that computes a probability based on the geodesic distance from user-provided scribbles for pixel classification; 2). Graph Cuts [133] that

[3] https://github.com/shelhamer/fcn.berkeleyvision.org
[4] https://bitbucket.org/deeplab/deeplab-public
[5] http://www.image-net.org

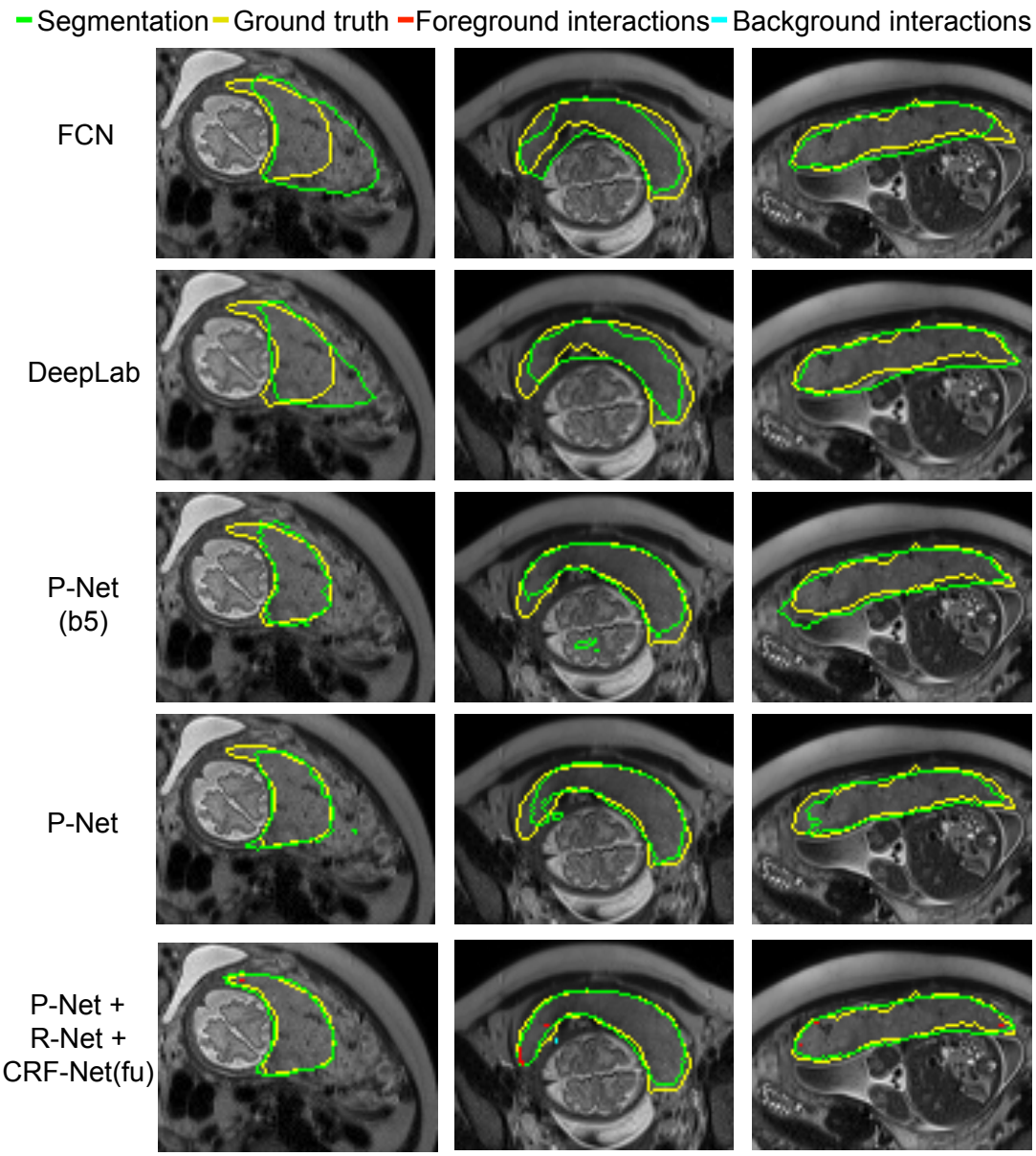


Figure 5.7: Initial automatic segmentation results of the placenta by P-Net. P-Net(b5) only uses the features from block 5 shown in Fig. 5.3 rather than the concatenated multi-scale features. Note the more accurate and detailed segmentation results of P-Net compared with FCN [123] and DeepLab [125]. The last row shows interactively refined results by DeepIGeoS.

models segmentation as a min-cut problem based on user interactions; 3). Random Walks [222] that assigns a pixel with a label based on the probability that a random walker reaches a foreground or background seed first, and 4). Slic-Seg [76] that uses Online Random Forests to learn from the scribbles and predict the labels of the remaining pixels. For quantitative evaluations of the segmentation results, the Dice

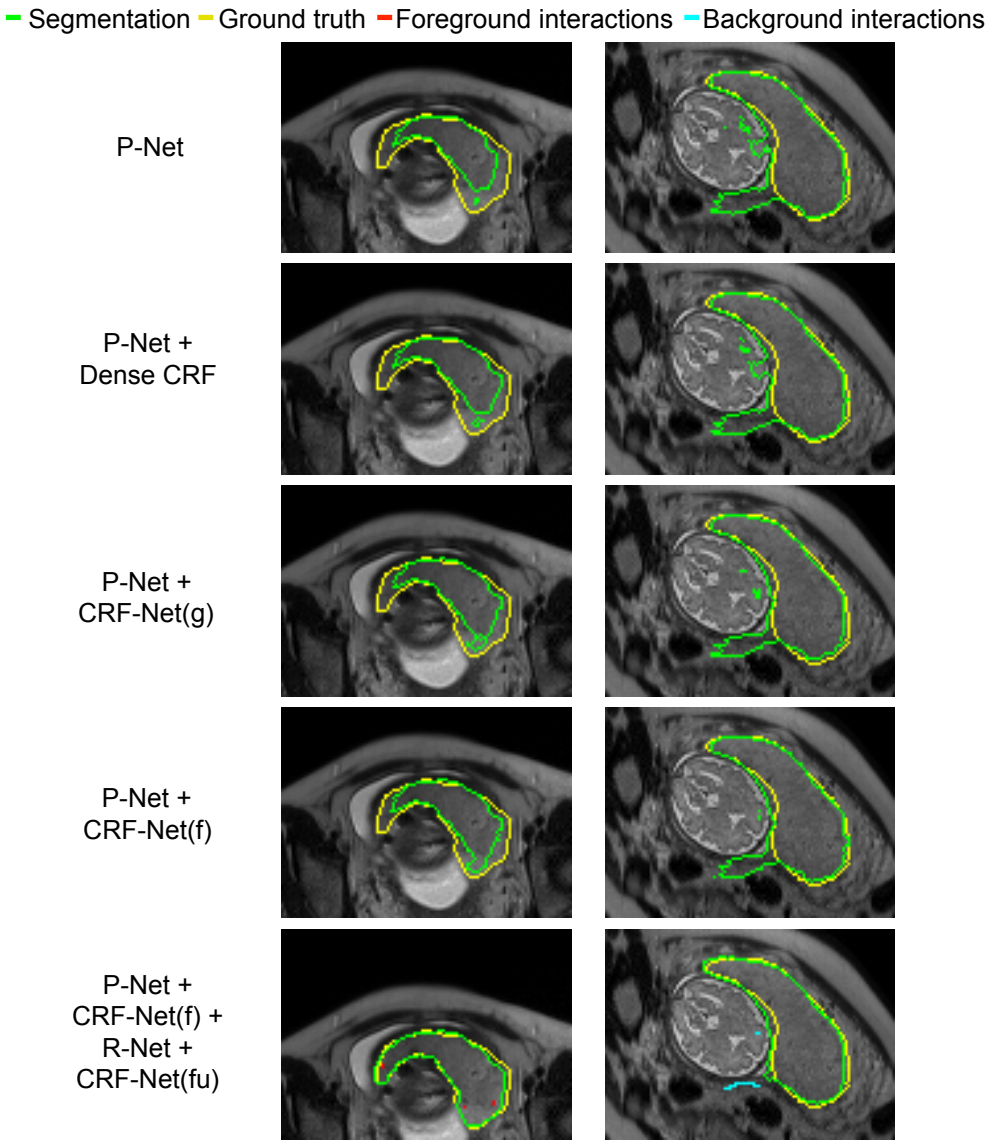


Figure 5.8: Visual comparison of placenta segmentation by P-Net with different CRFs. The last row shows interactively refined results by DeepIGeoS.

score defined in Eq. (3.12) and the ASSD defined in Eq. (4.5) were used. The Student's t-test was used to compute the p-value in order to see whether the results of two algorithms significantly differ from each other.

5.3.2 Automatic Segmentation by P-Net with CRF-Net(f)

Fig. 5.7 shows the automatic segmentation results obtained by different networks. It shows that FCN is able to capture the main region of the placenta. However, the segmentation results are blob-like with smooth boundaries. DeepLab is better than FCN,

Table 5.1: Quantitative comparison of placenta segmentation by different networks and CRFs. CRF-Net(g) [171] constrains pairwise potential as Gaussian functions. CRF-Net(f) is the proposed CRF that learns freeform pairwise potential functions. Significant improvement from P-Net (p-value$<$0.05) is shown in bold font.

Method	Dice(%)	ASSD(pixels)
FCN [123]	81.47±11.40	2.66±1.39
DeepLab [125]	83.38±9.53	2.20±0.84
P-Net(b5)	83.16±13.01	2.36±1.66
P-Net	84.78±11.74	2.09±1.53
P-Net + Dense CRF	84.90±12.05	2.05±1.59
P-Net + CRF-Net(g)	**85.44±12.50**	**1.98±1.46**
P-Net + CRF-Net(f)	**85.86±11.67**	**1.85±1.30**

but its blob-like results are similar to those of FCN. This is mainly due to the downsampling and upsampling procedure employed by these methods. In contrast, P-Net(b5) and P-Net obtain more detailed results. It can be observed that P-Net performs better than the other three networks. However, there are still some obvious mis-segmented regions by P-Net. A quantitative comparison of these networks based on all the testing data is shown in Table 5.1. P-Net achieves higher Dice score and lower ASSD compared with the other three networks. Compared with P-Net(b5), P-Net improves Dice from 83.16±13.01% to 84.78±11.74% and reduces the ASSD from 2.36±1.66 pixels to 2.09±1.53 pixels.

Based on the output of P-Net, three different CRFs are applied for spatial regularization respectively: Dense CRF, CRF-Net(g) with Gaussian pairwise potentials and CRF-Net(f) with freeform pairwise potentials. A visual comparison of them is shown in Fig. 5.8. In the first column, the placenta is under-segmented by P-Net. Dense CRF leads to very small improvements on the result. CRF-Net(g) and CRF-Net(f) improve the result by preserving more placenta regions, and the later shows a better segmentation. In the second column, P-Net obtains an over-segmentation of adjacent fetal brain and maternal tissues. Dense CRF does not improve the segmentation noticeably, but CRF-Net(g) and CRF-Net(f) remove more over-segmentated areas. CRF-Net(f) shows a better performance than the other two CRFs. The quantitative evaluation of these three CRFs is presented in Table 5.1, which shows Dense CRF leads to a result that is very close to that of P-Net (p-value $>$ 0.05), while the last two CRFs signifi-

Table 5.2: Quantitative comparison of different refinement methods for placenta segmentation. The initial segmentation is obtained by P-Net + CRF-Net(f). R-Net(Euc) uses Euclidean distance instead of geodesic distance. Significant improvement from R-Net (p-value$<$0.05) is shown in bold font.

Method	Dice(%)	ASSD(pixels)
Before refinement	85.86±11.67	1.85±1.30
Min-cut user-editing [136]	87.04±9.79	1.63±1.15
R-Net(Euc)	88.26±10.61	1.54±1.18
R-Net	88.76±5.56	1.31±0.60
R-Net(Euc) + CRF-Net(fu)	88.71±8.42	1.26±0.59
R-Net + CRF-Net(fu)	**89.31±5.33**	**1.22±0.55**

cantly improve the segmentation (p-value < 0.05). In addition, CRF-Net(f) is better than CRF-Net(g). Fig. 5.8 and Table 5.1 indicate that large mis-segmentation exists in some images, therefore R-Net with CRF-Net(fu) is used to refine the segmentation interactively in the following section.

5.3.3 Interactive Refinement by R-Net with CRF-Net(fu)

Fig. 5.9 shows examples of interactive refinement based on R-Net with CRF-Net(fu) that uses freeform pairwise potentials and employs user interactions as hard constraints. The first row in Fig. 5.9 shows initial automatic segmentation obtained by P-Net + CRF-Net(f). The user gives clicks/scribbles to indicate the foreground (red) or the background (cyan). The other rows in Fig. 5.9 show the results for five variants of refinement. These refinement methods correct most of the mis-segmented areas but perform at different levels in dealing with local details, as indicated by white arrows. Fig. 5.9 shows that R-Net with geodesic distance performs better than min-cut user-editing and R-Net(Euc) that uses Euclidean distance. CRF-Net(fu) can further improve the segmentation.

For a quantitative comparison, the segmentation accuracy after the first iteration of user refinement was measured, where the same initial segmentation and the same set of user interactions were used by the five refinement methods. The results are presented in Table 5.2, which shows that the combination of the proposed R-Net using geodesic distance and CRF-Net(fu) leads to more accurate segmentations than the other refinement methods with the same set of user interactions. The Dice score and ASSD of R-Net + CRF-Net(fu) are 89.31±5.33% and 1.22±0.55 pixels, respectively.

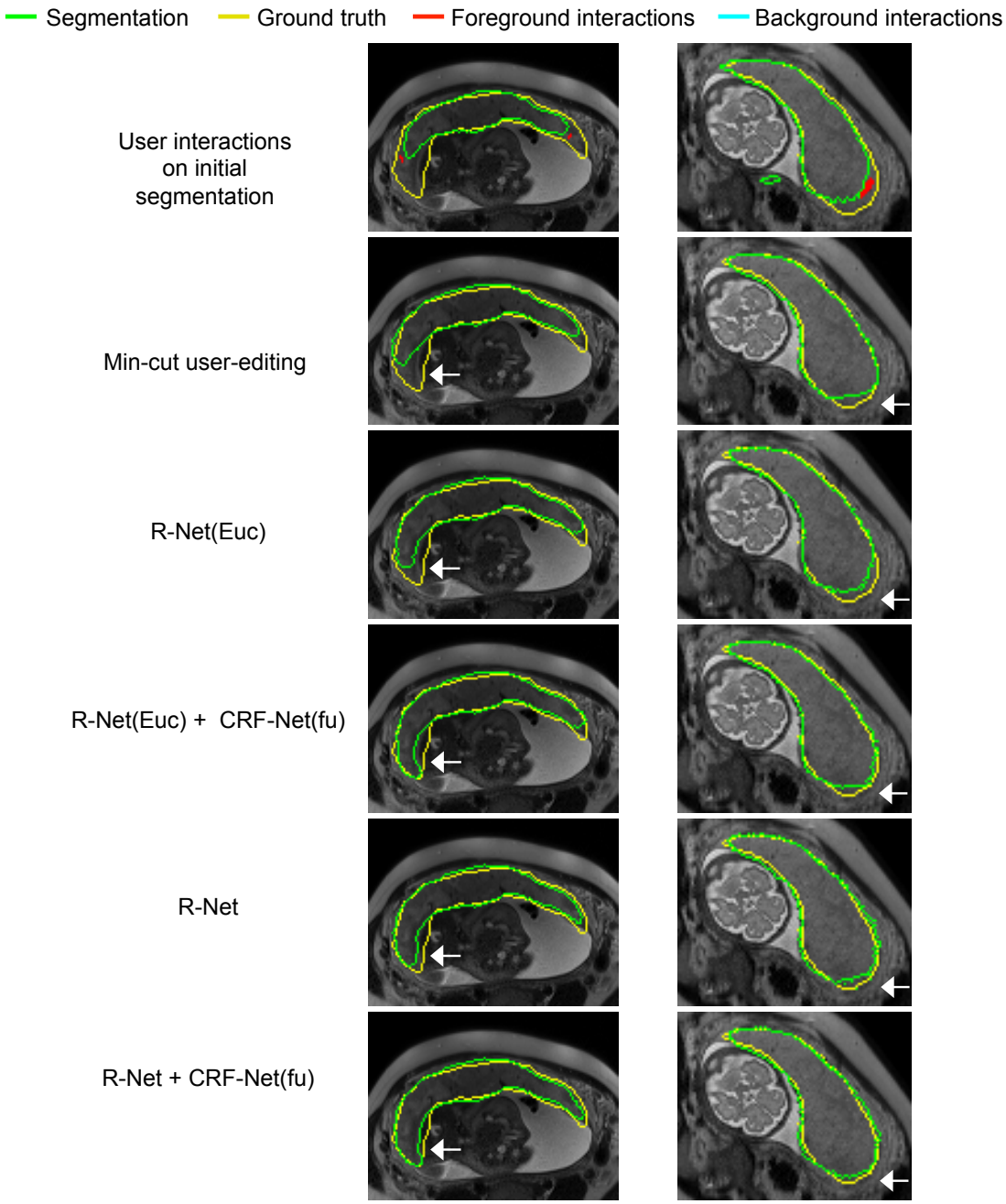


Figure 5.9: Visual comparison of different refinement methods for placenta segmentation. The first row shows the initial automatic segmentation obtained by P-Net + CRF-Net(f), on which user interactions are added for refinement. The remaining rows show refined results. R-Net(Euc) is a counterpart of the proposed R-Net, and it uses Euclidean distance.

5.3.4 Comparison with Other Interactive Methods

DeepIGeoS was compared with Geodesic Framework [221], Graph Cuts [133], Random Walks [222] and Slic-Seg [76] for placenta segmentation. A visual comparison is

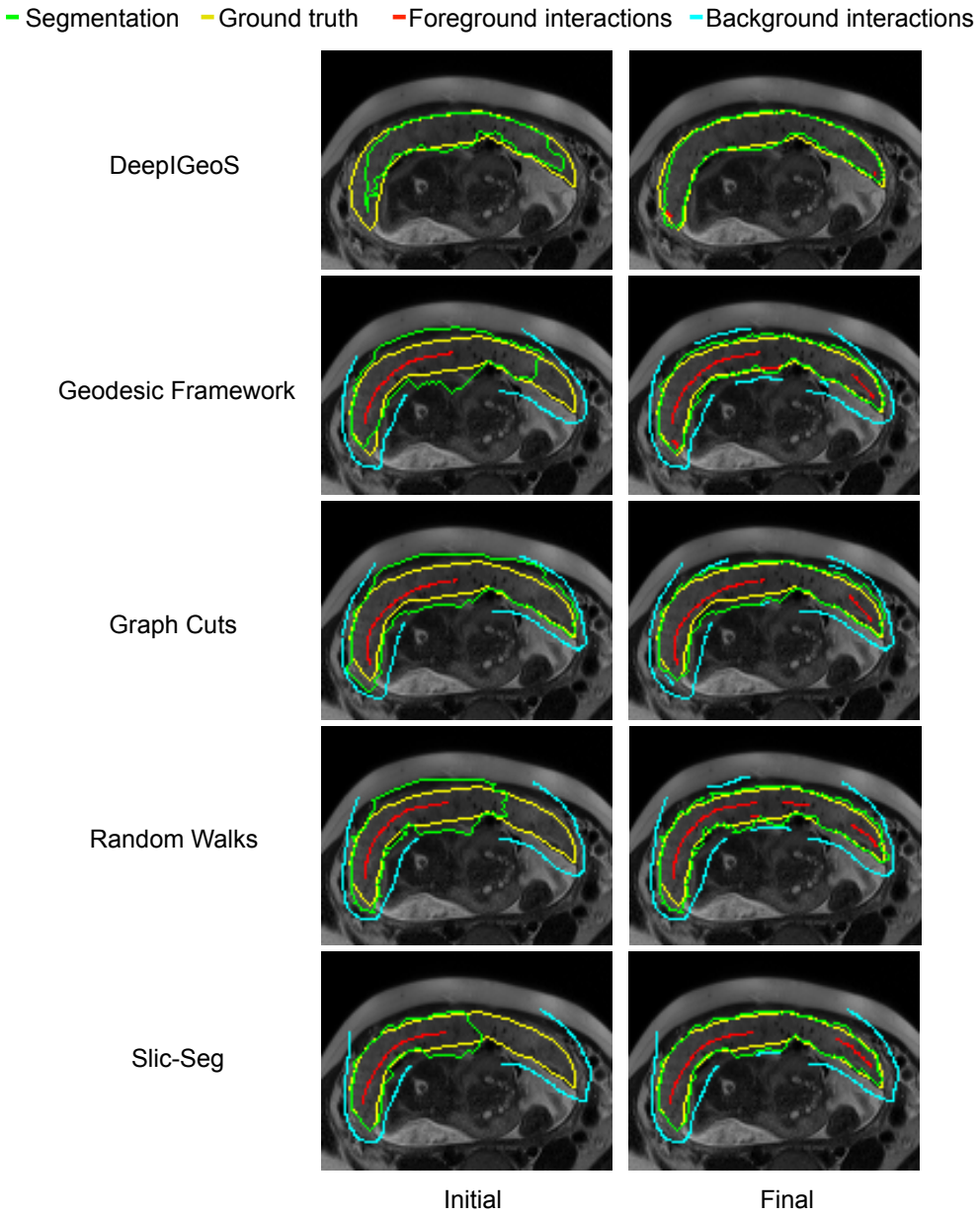


Figure 5.10: Visual comparison of DeepIGeoS and other interactive methods for placenta segmentation. The first column shows initial scribbles (except for DeepIGeoS) and the resulting segmentation. The second column shows final refined results with the entire set of scribbles. The user decided on the level of interactions required to achieve a visually acceptable result.

shown in Fig. 5.10. All these methods obtain an initial segmentation and then refine it. The first column shows the initial segmentation, where DeepIGeoS obtains a good result without user interactions while the other methods obtain worse results even with a large number of user interactions. The second column shows refined results, where

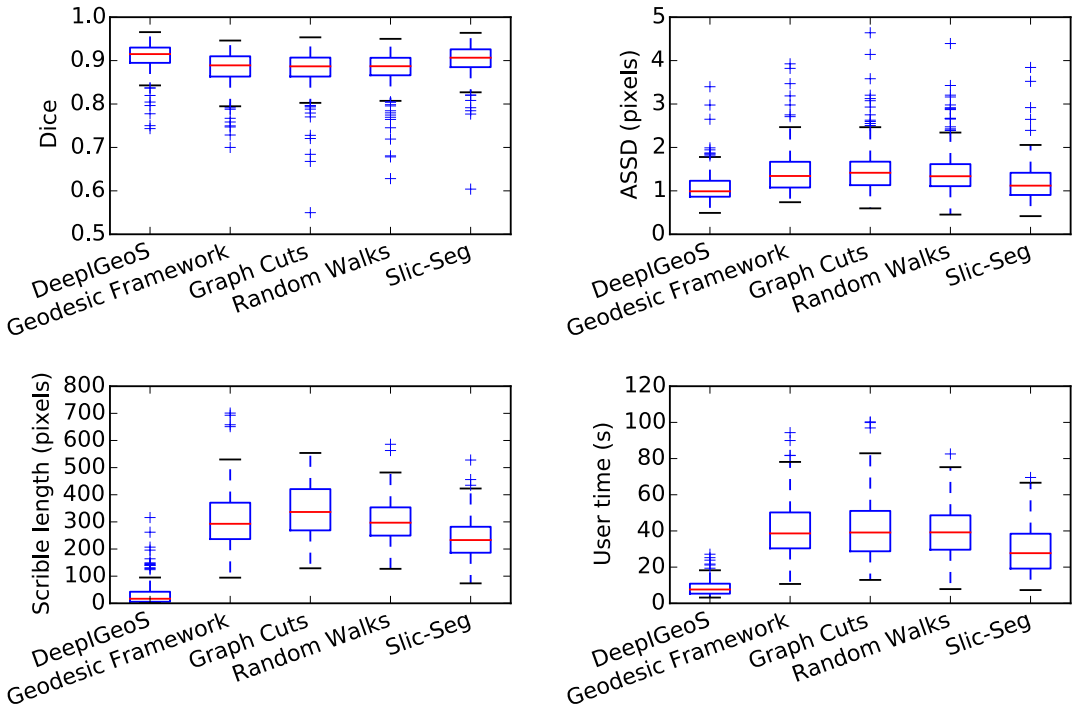


Figure 5.11: Quantitative comparison of placenta segmentation by different interactive methods in terms of Dice, ASSD, total interactions (scribble length) and user time.

DeepIGeoS only needs two short strokes to get an accurate segmentation, while the other methods additionally require more scribbles to get similar results. Two users (an Obstetrician and a Radiologist) were asked to use these methods to segment the placenta respectively. For each method, the segmentation of an image was refined until the user accepted the segmentation result. A quantitative comparison of these interactive methods is presented in Fig. 5.11. It shows that these interactive methods obtain similar accuracy for the final segmentation results, but DeepIgeoS needs fewer user interactions and less user time than the others.

5.4 Discussion and Conclusion

Differently from traditional interactive segmentation methods that require user inputs to get an initial segmentation, DeepIGeoS relies on a CNN to get the initial segmentation automatically and only requires the user to give interactions on mis-segmented areas. Therefore, it can considerably reduce the number of user interactions and user time. Though CNNs are the state-of-the-art automatic segmentation methods, the experiments have shown that the automatic methods provide a good starting point but

still need to be refined for higher accuracy. Experimental results demonstrate that the use of interactive refinement methods like DeepIGeoS is valuable and reasonable to achieve segmentation results that are satisfying for the user, while reducing the number of necessary interactions.

In chapter 3, the learning model was updated on the fly each time when new user interactions are given. Re-training the model during the interactive segmentation process is reasonable and efficient when ORFs are used and when the training set is a single image. However, it is inefficient for CNNs due to back-propagation and learning from a large dataset. Therefore, differently from Chapter 3 that re-trains the ORFs when new user interactions are given [74], the proposed R-Net is only trained once beforehand to deal with user interactions. Experimental results demonstrate this is enough to achieve good interactive efficiency and accurate segmentation results with only a small number of user interactions.

The proposed interactive segmentation framework can also be used to deal with other organs. In Appendix A, supplementary results of using DeepIGeoS for clavicle segmentation from chest radiographs are presented. The results also show that DeepIGeoS can achieve accurate segmentation efficiently and it requires noticeable fewer user interactions and less user time than traditional interactive segmentation methods. Appendix B demonstrates that DeepIGeoS can be extended to deal with 3D images, and shows that the 3D version of DeepIGeoS performs better than traditional interactive 3D segmentation methods such as GeoS [140] and ITK-SNAP [138]. It can also be extended to deal with a stack of motion-corrupted slices. Since the training dataset contains slices with different appearances, to segment one motion-corrupted volume, each slice can be segmented by the CNN independently and then a post-processing considering intra-volume consistency can be applied. For multiple volumes of the same patient, each volume can be first segmented independently in this way and the results provide an initialization for the co-segmentation framework presented in Chapter 4. In addition, extending DeepIGeoS to interactive multi-organ or multi-modal segmentation is also of interest.

In conclusion, this chapter presents a deep learning-based interactive framework

for placenta segmentation from fetal MR images. A P-Net is proposed to get an initial automatic segmentation and an R-Net is proposed to refine the result based on user interactions. The user interactions are transformed into geodesic distance maps and then integrated into the input of R-Net. This chapter also proposes a resolution-preserving network structure with dilated convolution for dense prediction, and extends the existing RNN-based CRF so that it can learn freeform pairwise potentials and take advantage of user-interactions as hard constraints. Segmentation results of placenta from fetal MR images show that the proposed method achieves better results than automatic CNNs. Compared with traditional interactive segmentation methods, it also obtains highly accurate results, but requires fewer user interactions and less user time. It can also be easily employed to deal with other segmentation tasks, as shown in Appendix A and B.

Chapter 6

Deep Interactive Segmentation with Image-specific Fine-tuning

6.1 Introduction

The work presented in this chapter is from my article published in TMI [223].

Chapter 5 has shown deep interactive segmentation with CNNs can achieve accurate segmentation with reduced user interactions compared with traditional interactive segmentation methods. Though DeepIGeoS achieves a high performance for placenta segmentation, it relies on a large number of annotated images for training. The model is trained to capture the representation of the placenta, therefore it cannot be used to segment other organs or unseen objects. For example, to segment the fetal lungs, some annotated images of fetal lungs are needed to train the CNN model. For medical images, annotations are often expensive to acquire as both expertise and time are needed to produce accurate annotations. This limits the performance of CNNs to segment objects for which annotations are not available at training time.

In addition, interactive segmentation often requires image-specific learning to deal with the large context variation among different images, but current CNNs are not adaptive to different test images as parameters of the model are learned from training images and then fixed during the testing, without image-specific adaptation. It has been shown that image-specific adaptation of a pre-trained Gaussian Mixture Model (GMM) helps to improve segmentation accuracy [224]. However, transitioning from simple

GMMs to powerful but complex CNNs in this context has not yet been demonstrated.

The aims of this chapter are two-fold: 1) to improve CNNs' ability to generalize to different organs so that the requirement of annotated data can be reduced for training and the model can deal with unseen object classes, and 2) to allow a pre-trained CNN model to be adaptive to a specific test image, which has a potential to improve the segmentation accuracy.

The contributions of this chapter are four-fold. First, this chapter proposes a novel deep learning-based framework for interactive 2D and 3D medical image segmentation by incorporating CNNs into a bounding box and scribble-based segmentation pipeline. Second, this chapter proposes to use image-specific fine-tuning to adapt a CNN model to each test image independently. The fine-tuning can be either unsupervised (without additional user interactions) or supervised where user-provided scribbles will guide the learning process. Third, this chapter proposes a weighted loss function considering network and interaction-based uncertainty during image-specific fine-tuning. Fourth, this chapter presents the first attempt to employ CNNs to segment previously unseen objects. The proposed framework is validated with 2D segmentation of multiple organs from fetal MR slices, where only two types of these organs are annotated for training; and 3D segmentation of brain tumor core (excluding edema) and whole brain tumor (including edema) from different MR sequences, where only tumor cores in one MR sequence are annotated for training.

6.2 Method

The proposed interactive segmentation framework using deep learning with image-specific fine-tuning is depicted in Fig. 6.1. It is referred to as BIFSeg. To deal with different (including previously unseen) objects in a unified framework, this chapter proposes to use a CNN that takes as input the content of a bounding box of one instance and gives a binary segmentation. In the testing stage, the bounding box is provided by the user, and the segmentation and the CNN are alternatively refined through unsupervised (without additional user interactions) or supervised (with user-provided scribbles) image-specific fine-tuning. The framework is general, flexible and can handle

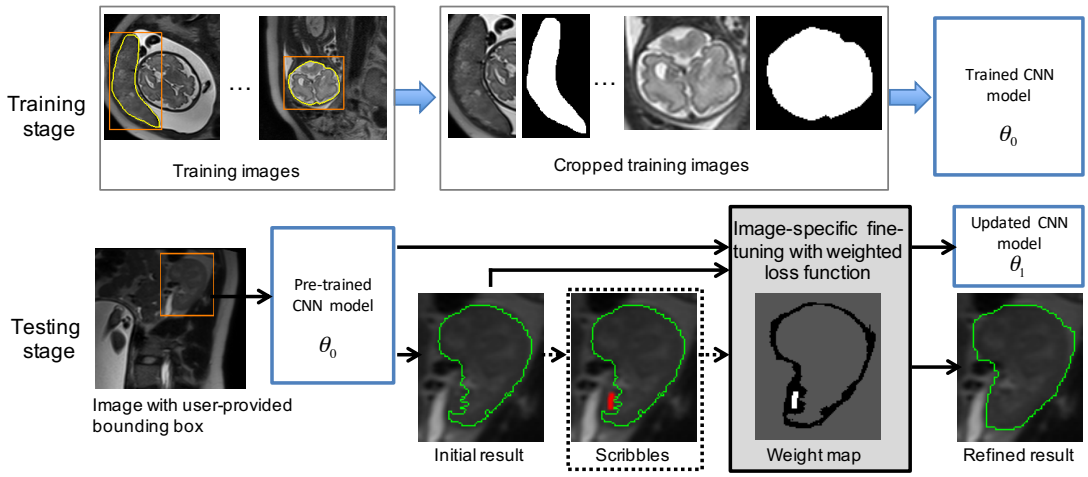


Figure 6.1: The proposed interactive segmentation framework (BIFSeg). 2D images are shown for examples. In the training stage, each instance is cropped with its bounding box, and the CNN model is trained for binary segmentation. In the testing stage, image-specific fine-tuning with optional scribbles and a weighted loss function is used. Note that the object class (e.g. a maternal kidney) in the test image may have not been present in the training set.

both 2D and 3D segmentations. In this chapter I choose to use the network structures proposed in Chapter 5. The contribution of BIFSeg is nonetheless largely different from DeepIGeoS in Chapter 5 as BIFSeg focuses on segmentation of previously unseen object classes and fine-tunes the CNN model on the fly for image-wise adaptation that can be guided by user interactions.

6.2.1 CNN Models

For 2D images, this chapter adopts the 2D P-Net (Fig. 5.3) for bounding box-based binary segmentation. To ensure efficient fine-tuning and fast response to user interactions, only parameters of the classifier (block 6) are fine-tuned. Thus, features in the concatenation layer for the test image can be stored before the fine-tuning.

For 3D images, this chapter extends the 2D P-Net (Fig. 5.3) with 3D convolutions. As shown in Fig. 6.2, the 3D network structure is similar to 2D P-Net. It consists of six blocks of layers. The first five blocks use convolution with dilation parameters 1, 2, 4, 8 and 16, respectively, so that they extract features at different scales. The first two blocks use convolution kernels of size $3 \times 3 \times 3$, and the following three blocks use convolution kernels of size $3 \times 3 \times 1$. This leads to an anisotropic receptive field $85 \times 85 \times 9$. Compared with slice-based networks, it employs 3D contexts. Compared

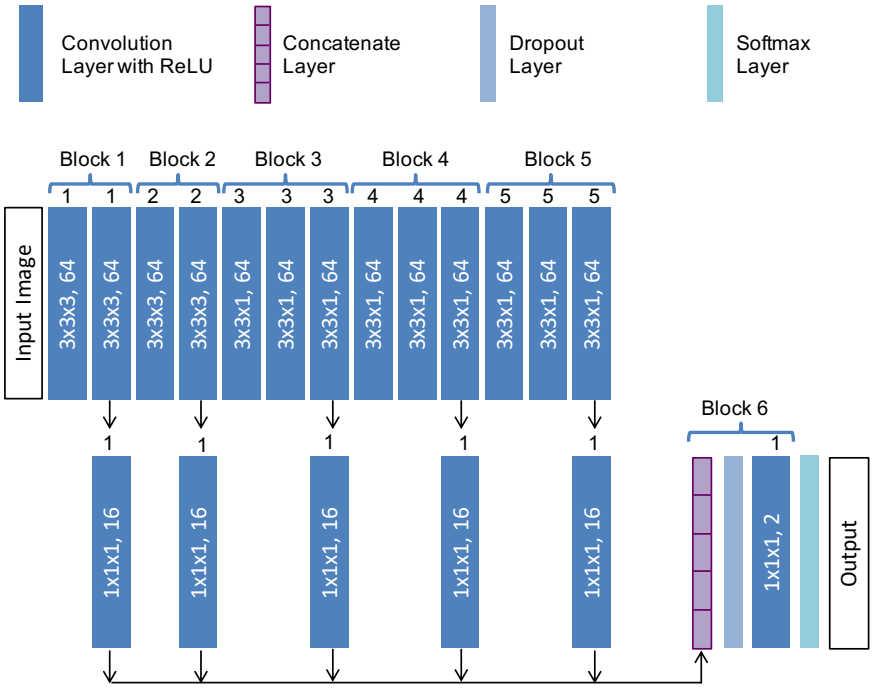


Figure 6.2: Proposed network with dilated convolution for 3D segmentation (PC-Net). The numbers in each dark blue box denote convolution kernel size and number of output channels. The number on the top denotes dilation parameter. For 2D segmentation in this Chapter, the P-Net proposed in Chapter 5 (Fig. 5.3) is used. During the image-specific fine-tuning process, the first five blocks in P-Net/PC-Net are fixed and only block 6 (the classifier) is fine-tuned.

with large isotropic 3D receptive fields [131], it has less memory consumption during inference [225]. Besides, anisotropic acquisition is often used in MR images. Features in blocks 1 to 5 are compressed by $1 \times 1 \times 1$ convolutions to save space and then the compressed features are fed into a concatenation layer. The concatenated features are used as the input of block 6 that serves as a classifier with $1 \times 1 \times 1$ convolutions. This 3D network with feature compression is referred to as PC-Net. Similarly to 2D P-Net, the first five blocks are fixed and only the classifier (block 6) is fine-tuned during the proposed image-specific fine-tuning process.

6.2.2 Training of CNNs

The training stage for 2D/3D segmentation is shown in the first row of Fig. 6.1. Consider a K-ary segmentation training set $T = \{(X_1, Y_1),\ (X_2, Y_2),\ ...\}$ where X_p is one training image and Y_p is the corresponding label map. The label set of T is $\{0,\ 1,\ 2,\ ...,\ K-1\}$ with 0 being the background label. Let N_k denote the number of

instances of the kth object type, so the total number of instances is $\hat{N} = \sum_k N_k$. Each image X_p can have instances of multiple object classes. Suppose the label of the qth instance in X_p is l_{pq}, Y_p is converted into a binary image Y_{pq} based on whether the value of each pixel in Y_p equals to l_{pq}. The bounding box B_{pq} of that training instance is automatically calculated based on Y_{pq} and expanded by a random margin in the range of 0 to 10 pixels/voxels. X_p and Y_{pq} are cropped based on B_{pq}. Thus, T is converted into a cropped set $\hat{T} = \{(\hat{X}_1, \hat{Y}_1),\ (\hat{X}_2, \hat{Y}_2),\ ...\}$ with size $\hat{N}$ and label set $\{0,\ 1\}$ where 1 is the label of the instance foreground and 0 the background. With $\hat{T}$, the CNN model (e.g, P-Net or PC-Net) is trained to extract the target from its bounding box, which is a binary segmentation problem irrespective of the object type. A cross entropy loss function is used for training.

6.2.3 Unsupervised and Supervised Image-specific Fine-tuning

In the testing stage, let $\hat{X}$ denote the sub-image inside a user-provided bounding box and $\hat{Y}$ be the target label of $\hat{X}$. The set of parameters of the trained CNN is θ. With the initial segmentation $\hat{Y}_0$ obtained by the trained CNN, the user may provide (i.e., supervised) or not provide (i.e., unsupervised) a set of scribbles to guide the update of $\hat{Y}_0$. Let S^f and S^b denote the scribbles for foreground and background, respectively, so the entire set of scribbles is $S = S^f \cup S^b$. Let s_i denote the user-provided label of a pixel in the scribbles, then $s_i = 1$ if $i \in S_f$ and $s_i = 0$ if $i \in S_b$. The proposed method minimizes an objective function that is similar to GrabCut [136] but it uses P-Net or PC-Net instead of a GMM:

$$\begin{aligned} &\arg\min_{\hat{Y},\theta} \left\{ E(\hat{Y}, \theta) = \sum_i \psi(\hat{y}_i | \hat{X}, \theta) + \lambda \sum_{i,j} \phi(\hat{y}_i, \hat{y}_j | \hat{X}) \right\} \\ &\text{subject to : } \hat{y}_i = s_i \quad \text{if } i \in S \end{aligned} \tag{6.1}$$

where $E(\hat{Y}, \theta)$ is constrained by user interactions if S is not empty. ψ and ϕ are the unary and pairwise energy terms, respectively. λ is the weight of ϕ.

An unconstrained optimization of an energy similar to E is used in [155] for weakly supervised learning. In that work, the energy was based on the probability and label map of all the images in a training set, which is a different task from this

work that focuses on a single test image. This chapter follows a typical choice of ϕ [133]:

$$\phi(\hat{y}_i, \hat{y}_j|\hat{X}) = [\hat{y}_i \neq \hat{y}_j]\exp\left(-\frac{(\hat{X}(i)-\hat{X}(j))^2}{2\sigma^2}\right)\cdot\frac{1}{d_{ij}} \tag{6.2}$$

where $[\cdot]$ is 1 if $\hat{y}_i \neq \hat{y}_j$ and 0 otherwise. d_{ij} is the Euclidean distance between pixel i and pixel j. σ controls the effect of intensity difference. ψ is defined as:

$$\psi(\hat{y}_i|\hat{X}, \theta) = -\log P(\hat{y}_i|\hat{X}, \theta) \tag{6.3}$$

$P(\hat{y}_i|\hat{X}, \theta)$ is the probability given by softmax output of the CNN. Let $p_i = P(\hat{y}_i = 1|\hat{X}, \theta)$ be the probability of pixel i belonging to the foreground, then:

$$\log P(\hat{y}_i|\hat{X}, \theta) = \hat{y}_i \log p_i + (1-\hat{y}_i)\log(1-p_i) \tag{6.4}$$

The optimization of Eq. (6.1) can be decomposed into steps that alternatively update the segmentation label $\hat{Y}$ and network parameters θ [155, 136]. In the label update step, the algorithm fixes θ and solves for $\hat{Y}$, and Eq. (6.1) becomes a CRF problem:

$$\begin{aligned} &\arg\min_{\hat{Y}} \left\{ E(\theta) = \sum_i \psi(\hat{y}_i|\hat{X}, \theta) + \lambda \sum_{i,j} \phi(\hat{y}_i, \hat{y}_j|\hat{X}) \right\} \\ &\text{subject to : } \hat{y}_i = s_i \quad \text{if } i \in S \end{aligned} \tag{6.5}$$

For implementation ease, the constrained optimization in Eq. (6.5) is converted to an unconstrained equivalent:

$$\arg\min_{\hat{Y}} \left\{ \sum_i \psi'(\hat{y}_i|\hat{X}, \theta) + \lambda \sum_{i,j} \phi(\hat{y}_i, \hat{y}_j|\hat{X}) \right\} \tag{6.6}$$

$$\psi'(\hat{y}_i|\hat{X},\theta) = \begin{cases} +\infty & \text{if } i \in S \text{ and } \hat{y}_i = s_i \\ 0 & \text{if } i \in S \text{ and } \hat{y}_i \neq s_i \\ -\log P(\hat{y}_i|\hat{X},\theta) & \text{otherwise} \end{cases} \tag{6.7}$$

Since θ and therefore ψ' are fixed, and ϕ is submodular, Eq. (6.6) can be solved by Graph Cuts [133]. In the network update step, the algorithm fixes $\hat{Y}$ and solves for θ:

$$\begin{aligned} &\arg\min_{\theta} \left\{ E(\hat{Y}) = \sum_i \psi(\hat{y}_i|\hat{X},\theta) \right\} \\ &\text{subject to : } \hat{y}_i = s_i \quad \text{if } i \in S \end{aligned} \tag{6.8}$$

Thanks to the constrained optimization in Eq. (6.5), the label update step necessarily leads to $\hat{y}_i = s_i$ for $i \in S$. Eq. (6.8) can be treated as an unconstrained optimization:

$$\arg\min_{\theta} \left\{ -\sum_i \Big(\hat{y}_i \log p_i + (1-\hat{y}_i)\log(1-p_i) \Big) \right\} \tag{6.9}$$

6.2.4 Weighted Loss Function during Network Update Step

During the network update step, the CNN is fine-tuned to fit the current segmentation $\hat{Y}$. Compared with a standard learning process that treats all the pixels equally, this chapter proposes to weight different kind of pixels considering their confidence. First, user-provided scribbles have much higher confidence than the other pixels, and they should have a higher effect on the loss function, leading to a weighted version of Eq. (6.3):

$$\psi(\hat{y}_i|\hat{X},\theta) = -w(i)\log P(\hat{y}_i|\hat{X},\theta) \tag{6.10}$$

$$w(i) = \begin{cases} \omega & \text{if } i \in S \\ 1 & \text{otherwise} \end{cases} \tag{6.11}$$

where $\omega \geq 1$ is the weight associated with scribbles. ψ defined in Eq. (6.10) allows Eq. (6.5) to remain unchanged for the label update step. In the network update step, Eq. (6.9) becomes:

$$\arg\min_{\theta} \left\{ -\sum_{i} w(i) \Big(\hat{y}_i \log p_i + (1-\hat{y}_i) \log(1-p_i) \Big) \right\} \tag{6.12}$$

Note that the energy optimization problem of Eq. (6.1) remains well-posed with Eq. (6.10), (6.11), and (6.12).

Second, $\hat{Y}$ may contain mis-classified pixels that can mis-lead the network update process. To address this problem, this chapter proposes to fine-tune the network by ignoring pixels with high uncertainty (low confidence) in the test image. The uncertainty includes network-based uncertainty and scribble-based uncertainty. The network-based uncertainty is based on the network's softmax output. Since $\hat{y}_i$ is highly uncertain (has low confidence) if p_i is close to 0.5, this chapter defines the set of pixels with high network-based uncertainty as $U_p = \{i | t_0 < p_i < t_1\}$ where t_0 and t_1 are the lower and higher threshold values of foreground probability, respectively. The scribble-based uncertainty is based on the geodesic distance to scribbles. Let $G(i, S^f)$ and $G(i, S^b)$ denote the geodesic distance [140] from pixel i to S^f and S^b, respectively. Since the scribbles are drawn on mis-segmented areas for refinement, it is likely that pixels close to S have been incorrectly labeled by the initial segmentation. Let ε be a threshold value for the geodesic distance. This chapter defines the set of pixels with high scribble-based uncertainty as $U_s = U_s^f \cup U_s^b$ where $U_s^f = \{i | i \notin S, G(i, S^f) < \varepsilon, \hat{y}_i = 0\}$, $U_s^b = \{i | i \notin S, G(i, S^b) < \varepsilon, \hat{y}_i = 1\}$. Therefore, a full version of the weighting function is (an example is shown in Fig. 6.3):

$$w(i) = \begin{cases} \omega & \text{if } i \in S \\ 0 & \text{if } i \in U_p \cup U_s \\ 1 & \text{otherwise} \end{cases} \tag{6.13}$$

The new definition of $w(i)$ is well motivated in the network update step. However, in

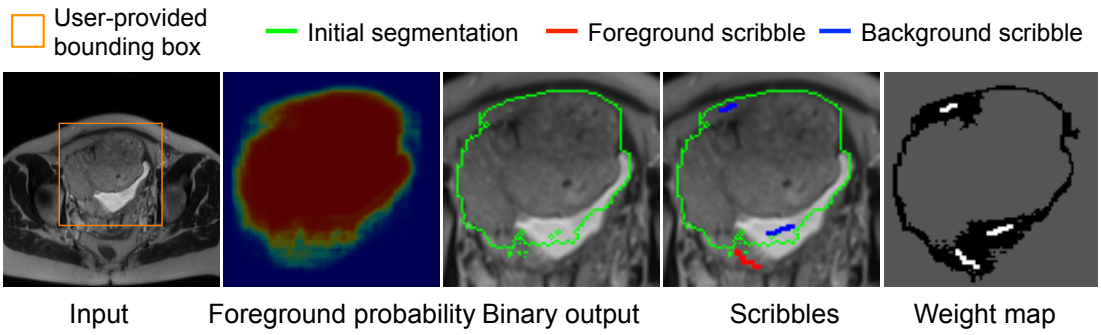


Figure 6.3: An example of weight map for image-specific fine-tuning. In the last image, the weight is 0 for pixels with high uncertainty (black), ω for scribbles (white), and 1 for the remaining pixels (gray).

the label update step, introducing zero unary weights in Eq. (6.5) would make the label update of corresponding pixels entirely driven by the pairwise potentials. Therefore, this chapter choose to keep Eq. (6.5) unchanged.

6.2.5 Implementation Details

The Caffe[1] [219] deep learning library was used to implement the P-Net and PC-Net. The training process was done via one node of the Emerald cluster[2] with two 8-core E5-2623v3 Intel Haswells, a K80 NVIDIA GPU and 128GB memory. The training of CNNs used stochastic gradient decent, with momentum 0.9, batch size 1, weight decay 5×10^{-4}, maximal number of iterations 60k, initial learning 10^{-3} that was halved every 5k iterations. For each application, the images in each modality were normalized by the mean value and standard deviation of the training images. During training, the bounding box for each object was automatically generated based on the ground truth label with a random margin in the range of 0 to 10 pixels/voxels. For convolutions at the border of an input, spatial reflection padding [226] was used to ensure that the output had the same size as the input.

For the testing with user interactions, the trained CNN models were deployed to a MacBook Pro (OS X 10.9.5) with 16GB RAM, an Intel Core i7 CPU running at 2.5GHz and an NVIDIA GeForce GT 750M GPU. A Matlab GUI and a PyQt GUI were used for user interactions on 2D and 3D images, respectively. The bounding box

[1] `http://caffe.berkeleyvision.org`
[2] `http://www.ses.ac.uk/high-performance-computing/emerald`

was provided by the user. For image-specific fine-tuning, $\hat{Y}$ and θ were alternatively updated for four iterations. In each network update step, the learning process used a learning rate 10^{-2} and iteration number 20. A grid search with the training data was used to get proper values of λ, σ, t_0, t_1, ε and ω. Their numerical values are listed in the specific experiments sections 6.3.2 and 6.3.3.

6.3 Experiments and Results

The proposed framework was validated with two applications: 2D segmentation of multiple organs from fetal MR images and 3D segmentation of brain tumors from contrast enhanced T1-weighted (T1c) and Fluid-attenuated Inversion Recovery (FLAIR) images. For both applications, the experiments additionally investigated the segmentation performance on previously unseen objects that had not been present in the training set.

6.3.1 Comparison Methods and Evaluation Metrics

To investigate the performance of different networks with the same bounding box, this chapter compares P-Net with FCN [123] and U-Net [124] for 2D images, and compares PC-Net with DeepMedic [128] and HighRes3DNet [131] for 3D images[3]. The original DeepMedic works on multiple modalities, and this chapter adapts it to work on a single modality. All these methods were evaluated on the laptop during the testing except for HighRes3DNet that was run on the cluster due to the laptop's limited GPU memory. To validate the proposed unsupervised/supervised image-specific fine-tuning, this chapter compares BIFSeg with 1) the initial output of P-Net/PC-Net, 2) post-processing the initial output with a CRF (using user interactions as hard constraints if they are given), and 3) image-specific fine-tuning based on Eq. (6.1) with $w(i) = 1$ for all the pixels, which is referred to as BIFSeg(-w).

BIFSeg was also compared with other interactive segmentation methods: GrabCut [136], Slic-Seg [76] and Random Walks [135] for 2D segmentation, and GeoS [140], GrowCut [141] and 3D GrabCut [228] for 3D segmentation. The 2D/3D GrabCut used the same bounding box as used by BIFSeg, and they used 3 and 5 com-

[3]DeepMedic and HighRes3DNet were implemented in NiftyNet [227] `http://niftynet.io`

ponents for the foreground and background GMMs, respectively. Slic-Seg, Random Walks, GeoS and GrowCut required scribbles without a bounding box for segmentation. The segmentation results by an Obstetrician and a Radiologist were used for evaluation. For each method, each user provided scribbles to update the result multiple times until the user accepted it as the final segmentation. The Dice score between a segmentation and the ground truth defined in Eq. (3.12) was used for quantitative evaluations. The p-value between different methods was computed by the Student's t-test.

6.3.2 2D Segmentation of Multiple Organs from Fetal MR Images

6.3.2.1 Data

Stacks of T2-weighted MR images from 18 pregnant women in the second trimester were acquired by SSFSE with pixel size 0.74 to 1.58 mm and inter-slice spacing 3 to 4 mm. Due to the large inter-slice spacing and inter-slice motion, interactive 2D segmentation is more suitable than direct 3D segmentation [76]. The placenta and fetal brain from ten volumes (356 slices) were used for training. The other eight volumes (318 slices) were used for testing. From the test images, this chapter aims to segment the placenta, fetal brain, and previously unseen fetal lungs and maternal kidneys. Manual segmentations by a Radiologist were used as the ground truth. P-Net was used for this segmentation task. To deal with organs at different scales, the input of P-Net was resized so that the minimal value of width and height was 128 pixels. Parameter setting was $\lambda = 3.0$, $\sigma = 0.1$, $t_0 = 0.2$, $t_1 = 0.7$, $\varepsilon = 0.2$, $\omega = 5.0$ based on a grid search performed on the training data.

6.3.2.2 Initial Segmentation based on P-Net

Fig. 6.4 shows the segmentation of different organs from fetal MR images with user-provided bounding boxes. The first row presents the bounding box for each target organ. The other rows show the results of GrabCut and three different networks. It can be observed that GrabCut achieves a poor segmentation except for the fetal brain where there is a good contrast between the target and the background. For the placenta and fetal brain, FCN, U-Net and P-Net achieve visually similar results that are close to

Table 6.1: Quantitative comparison of initial segmentation of fetal MR images from a bounding box. $\wedge$ denotes previously unseen objects. In each row, bold font denotes the best value. * denotes p-value < 0.05 compared with the others.

		FCN	U-Net	P-Net	GrabCut
Dice (%)	Placenta	**85.31±8.73**	82.86±9.85	84.57±8.37	62.90±12.79
	Fetal brain	**89.53±3.91**	89.19±5.09	89.44±6.45	83.86±14.33
	Fetal lungs$^\wedge$	81.68±5.95	80.64±6.10	**83.59±6.42***	63.99±15.86
	Maternal kidneys$^\wedge$	83.58±5.48	75.20±11.23	**85.29±5.08***	73.85±7.77
Machine time (s)		**0.11±0.04***	0.24±0.07	0.16±0.05	1.62±0.42

the ground truth. However, for fetal lungs and maternal kidneys that are previously unseen in the training set, FCN and U-Net lead to a large region of under-segmentation. In contrast, P-Net performs noticeably better than FCN and U-Net when dealing with these two unseen objects. A quantitative evaluation of these methods is listed in Table 6.1. It shows that P-Net achieves the best accuracy for unseen fetal lungs and maternal kidneys with average machine time 0.16s.

6.3.2.3 Unsupervised Image-specific Fine-tuning

For unsupervised refinement, the initial segmentation result obtained by P-Net was refined by CRF, BIFSeg(-w) and BIFSeg without additional scribbles, respectively. The results are shown in Fig. 6.5. The second to fourth rows show the foreground probability given by P-Net before and after the fine-tuning. In the second row, the initial output of P-Net has a probability around 0.5 for many pixels, which indicates a high uncertainty. After image-specific fine-tuning, most pixels in the outputs of BIFSeg(-w) and BIFSeg have a probability close to 0.0 or 1.0. The remaining rows show the segmentations by P-Net and the three refinement methods, respectively. The visual comparison shows that BIFSeg performs better than P-Net + CRF and BIFSeg(-w). Quantitative measurements are presented in Table 6.2. It shows that BIFSeg achieves a larger improvement of accuracy from the initial segmentation when compared with the use of CRF or BIFSeg(-w). In this 2D case, BIFSeg takes 0.72s in average for unsupervised image-specific fine-tuning.

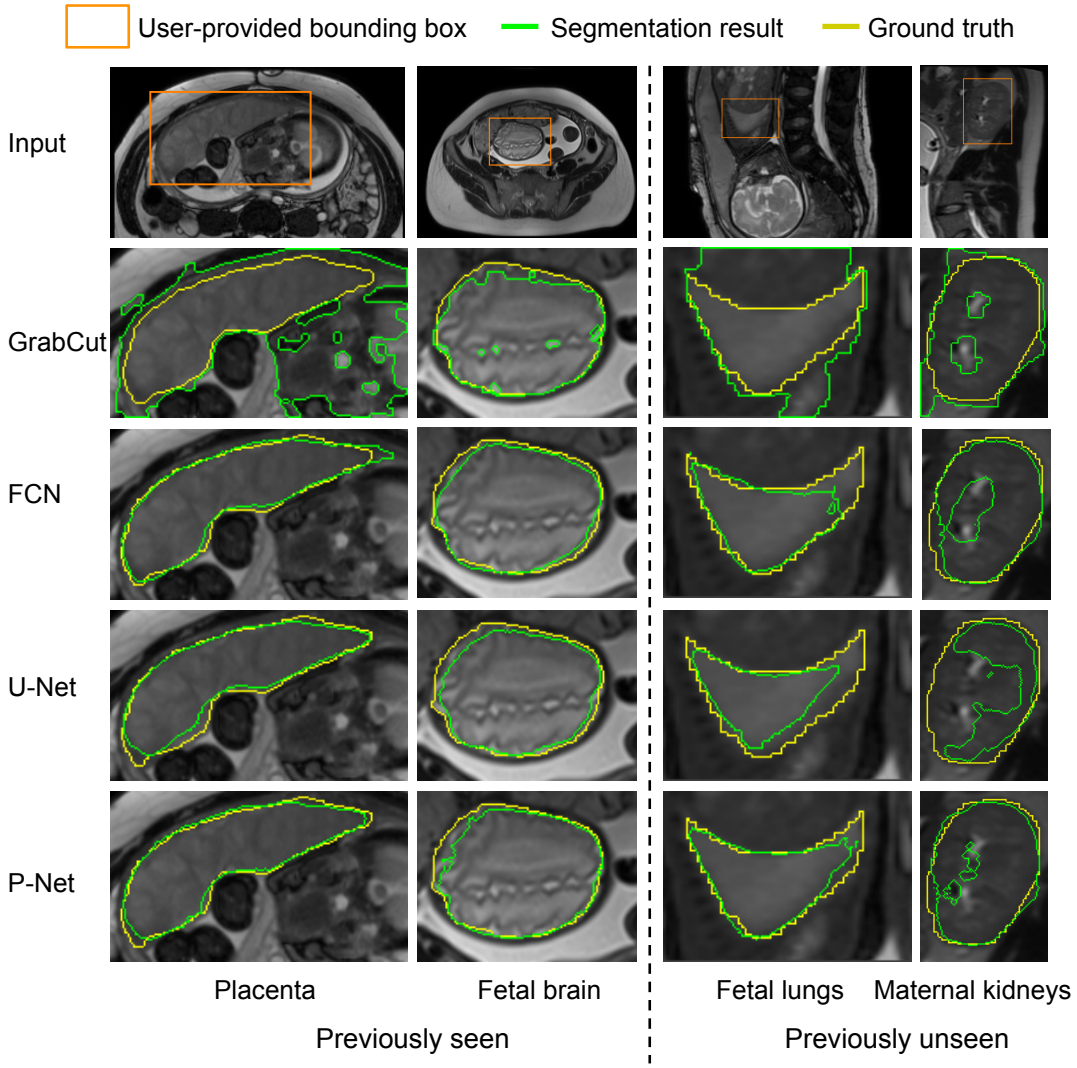


Figure 6.4: Visual comparison of initial segmentation of multiple organs from fetal MR images with a bounding box. GrabCut learns from a single image and the other methods learn from a training set. All the methods use the same bounding box for each test instance. Note that fetal lungs and maternal kidneys are previously unseen objects but P-Net works well on them.

6.3.2.4 Supervised Image-specific Fine-tuning

Fig. 6.6 shows examples of supervised refinement with additional scribbles. The second row shows the initial segmentation given by P-Net. In the third row, red and blue scribbles are drawn in mis-segmented regions to label the corresponding pixels as the foreground and background, respectively. The same initial segmentation and scribbles are used for P-Net + CRF, BIFSeg(-w) and BIFSeg. All these methods improve the segmentation. However, some large mis-segmentations can still be observed for P-Net

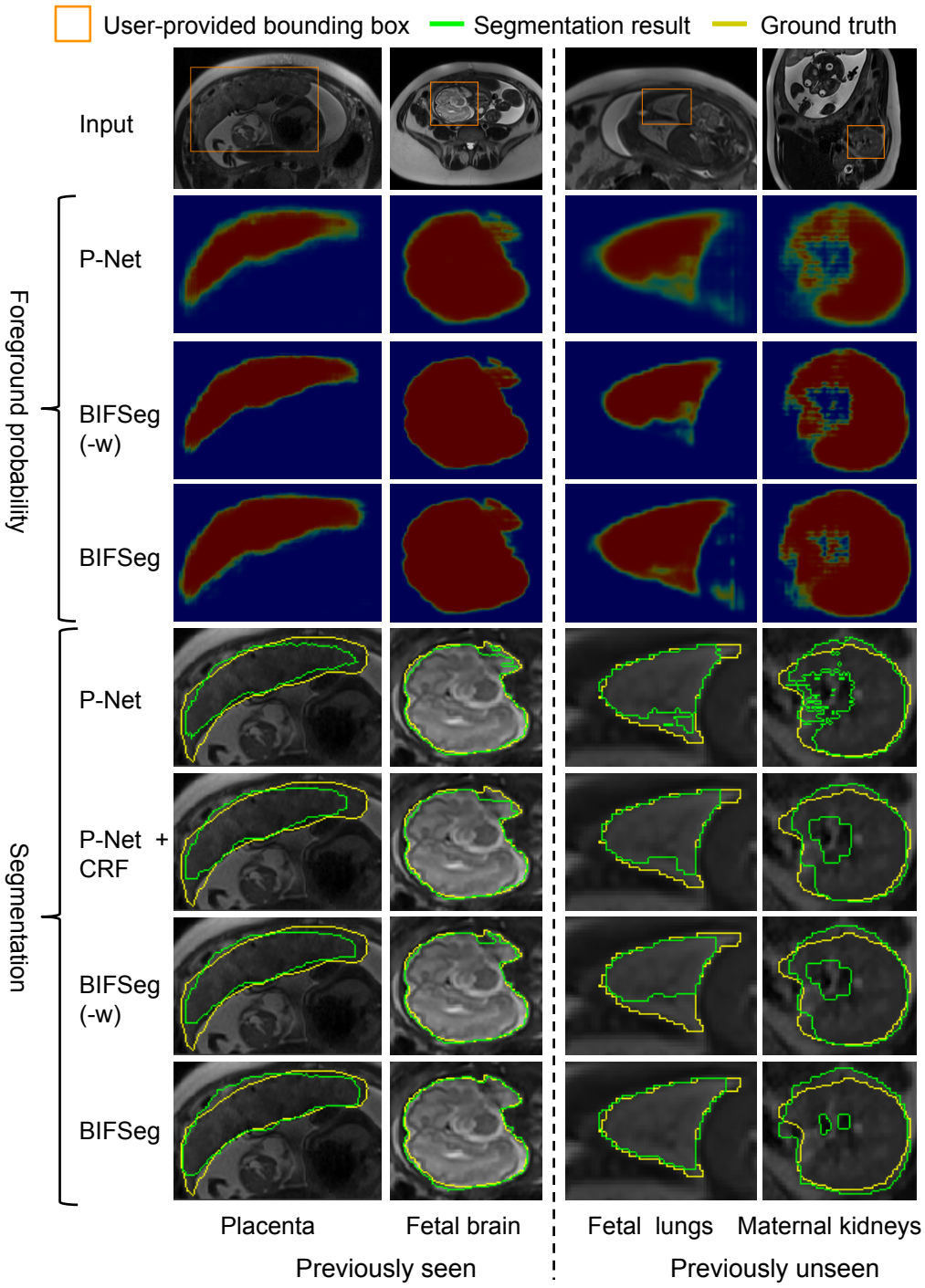


Figure 6.5: Visual comparison of P-Net and three unsupervised refinement methods without additional scribbles for segmentation of fetal MR images. The foreground probability is visualized by heatmap.

+ CRF and BIFSeg(-w). In contrast, BIFSeg achieves better results with the same set of scribbles. For a quantitative comparison, I measured the segmentation accuracy after a single round of refinement using the same set of scribbles. The result is shown

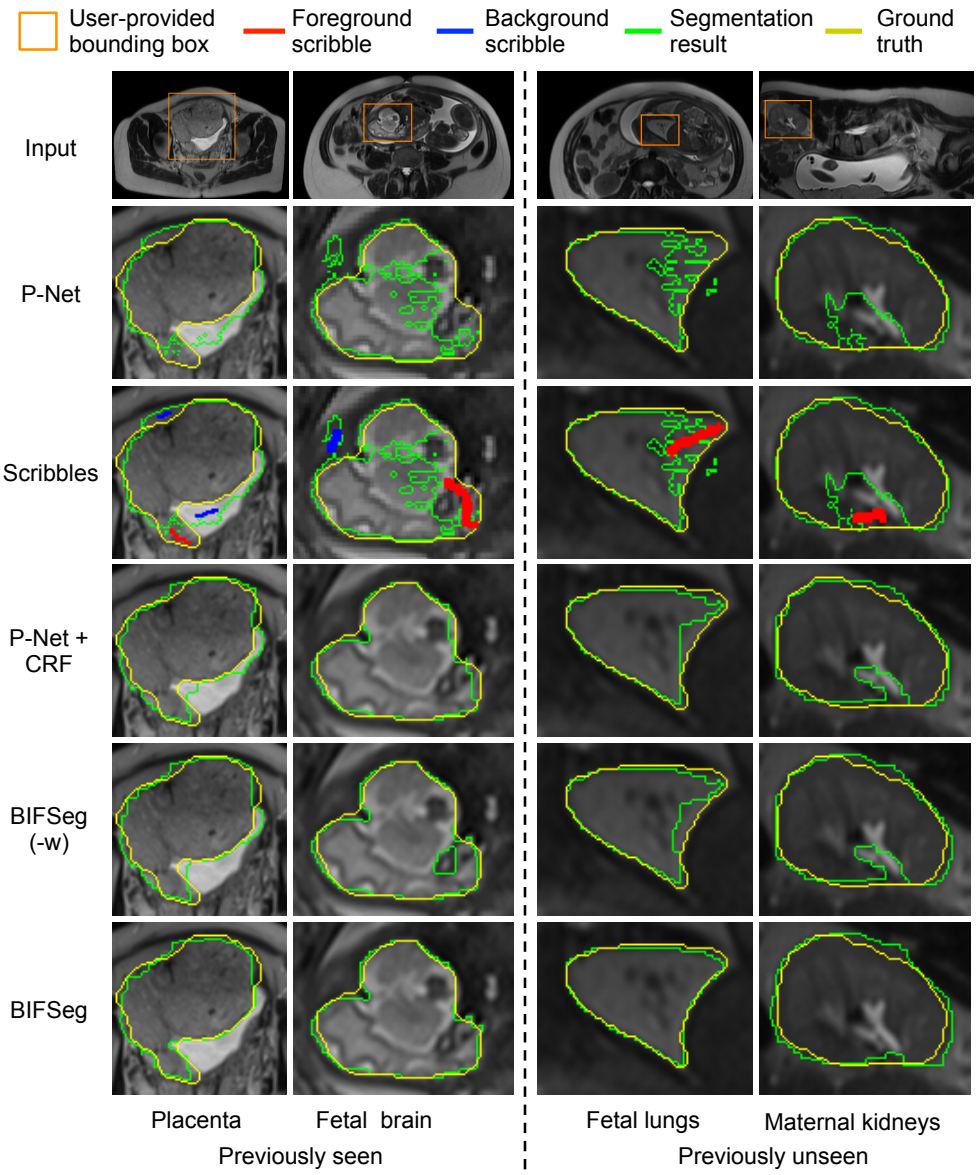


Figure 6.6: Visual comparison of P-Net and three supervised refinement methods for segmentation of fetal MR images. The same initial segmentation and scribbles are used for P-Net + CRF, BIFSeg(-w) and BIFSeg.

in Table 6.3. BIFSeg achieves significantly better accuracy (p-value < 0.05) for the placenta, and previously unseen fetal lungs and maternal kidneys compared with P-Net + CRF and BIFSeg(-w).

6.3.2.5 Comparison with other interactive methods

The two users (an Obstetrician and a Radiologist) used Slic-Seg [76], GrabCut [136], Random Walks [135] and BIFSeg for the fetal MR image segmentation tasks respec-

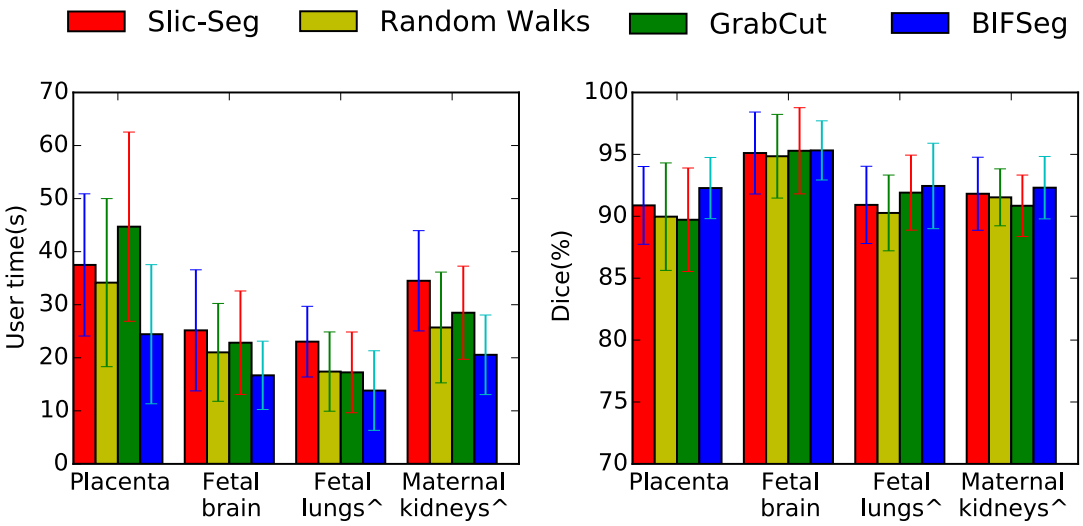


Figure 6.7: User time and Dice score of different interactive methods for segmentation of fetal MR images. $^\wedge$ denotes previously unseen objects for BIFSeg.

Table 6.2: Quantitative comparison of P-Net and three unsupervised refinement methods without additional scribbles for segmentation of fetal MR images. $\wedge$ denotes previously unseen objects. In each row, bold font denotes the best value. * denotes p-value $<$ 0.05 compared with the others.

		P-Net	P-Net+CRF	BIFSeg(-w)	BIFSeg
Dice (%)	Placenta	84.57±8.37	84.87±8.14	82.74±10.91	**86.41±7.50***
	Fetal brain	89.44±6.45	89.55±6.52	89.09±8.08	**90.39±6.44**
	Fetal lungs$^\wedge$	83.59±6.42	83.87±6.52	82.17±8.87	**85.35±5.88***
	Maternal kidneys$^\wedge$	85.29±5.08	85.45±5.21	84.61±6.21	**86.33±4.28***
Additional machine time (s)		-	**0.02±0.01***	0.71±0.12	0.72±0.12

tively. For each image, the user implemented the segmentation interactively until the result was accepted by the user. The user time and final accuracy of these methods are

Table 6.3: Quantitative comparison of different supervised refinement methods for segmentation of fetal MR images. P-Net gives an initial segmentation, and the last three columns show refinement results with additional scribbles. $\wedge$ denotes previously unseen objects. In each row, bold font denotes the best value. * denotes p-value $<$ 0.05 compared with the others.

		P-Net	P-Net+CRF	BIFSeg(-w)	BIFSeg
Dice (%)	Placenta	84.57±8.37	88.64±5.84	89.79±4.60	**91.93±2.79***
	Fetal brain	89.44±6.45	94.04±4.72	95.31±3.39	**95.58±1.94**
	Fetal lungs$^\wedge$	83.59±6.42	88.92±3.87	89.21±2.95	**91.71±3.18***
	Maternal kidneys$^\wedge$	85.29±5.08	87.51±4.53	87.78±4.46	**89.37±2.31***
Additional machine time (s)		-	**0.02±0.01***	0.72±0.11	0.74±0.12

presented in Fig. 6.7. It shows that BIFSeg takes noticeably less user time with similar or higher accuracy compared with the other three interactive segmentation methods.

6.3.3 3D Segmentation of Brain Tumors from T1c and FLAIR Images

6.3.3.1 Clinical Background and Data

Gliomas are the most common brain tumors in adults with little improvement in treatment effectiveness despite considerable research works [229]. With the development of medical imaging, brain tumors can be imaged by different MR protocols with different contrasts. For example, T1-weighted images highlight enhancing part of the tumor and FLAIR acquisitions highlight the peritumoral edema. Segmentation of brain tumors can provide better volumetric measurements and therefore has enormous potential value for improved diagnosis, treatment planning, and follow-up of individual patients. However, automatic brain tumor segmentation remains technically challenging because 1) the size, shape, and localization of brain tumors have considerable variations among patients; 2) the boundaries between adjacent structures are often ambiguous.

To validate the proposed method with 3D brain tumor images, I used the 2015 Brain Tumor Segmentation Challenge (BRATS) training set [229]. The ground truth were manually delineated by experts. This dataset was collected from 274 cases with multiple modalities with different contrasts. T1c highlights the tumor without peritumoral edema, designated "tumor core" as per [229]. FLAIR highlights the tumor with peritumoral edema, designated "whole tumor" as per [229]. This chapter investigates interactive segmentation of tumor cores from T1c images and whole tumors from FLAIR images, which is different from previous works on automatic multi-label and multi-modal segmentation [230, 128]. For tumor core segmentation, I randomly selected 249 T1c volumes as the training set and used the remaining 25 T1c volumes as the testing set. Additionally, to investigate dealing with unseen objects, I employed such trained CNNs to segment whole tumors in the corresponding FLAIR images of these 25 volumes that were not present in the training set. All these images had been

Table 6.4: Dice score of initial segmentation of 3D brain tumors from a bounding box. All the methods use the same bounding box for each test image. $\wedge$ denotes unseen objects. In each row, bold font denotes the best value. * denotes p-value < 0.05 compared with the others.

	DeepMedic	HighRes3DNet	PC-Net	GrabCut
Tumor core	76.68±11.83	**83.45±7.87**	82.66±7.78	69.24±19.20
Whole tumor$^\wedge$	**84.04±8.50**	75.60±8.97	83.52±8.76	78.39±18.66

skull-stripped and resampled to isotropic 1mm^3 resolution. To deal with 3D tumor cores and whole tumors at different scales, the cropped image region inside a bounding box was resized so that its maximal value of width, height and depth is 80. Parameter setting was $\lambda = 10.0$, $\sigma = 0.1$, $t_0 = 0.2$, $t_1 = 0.6$, $\varepsilon = 0.2$, $\omega = 5.0$ based on a grid search with the training data.

6.3.3.2 Initial Segmentation based on PC-Net

Fig. 6.8(a) shows an initial result of tumor core segmentation from T1c with a user-provided bounding box. Since the central region of the tumor has a low intensity close to that of the background, 3D GrabCut has a poor performance with under-segmentations. DeepMedic leads to some over-segmentations. HighRes3DNet and PC-Net obtain similar results, but PC-Net is less complex and has a lower memory consumption. Fig. 6.8(b) shows the initial segmentation result of a previously unseen whole tumor from FLAIR. 3D GrabCut fails to get a high accuracy due to intensity inconsistency in the tumor region. The CNNs outperform 3D GrabCut, and DeepMedic and PC-Net perform better than HighRes3DNet. A quantitative comparison is presented in Table 6.4. It shows that the performance of DeepMedic is low for T1c but high for FLAIR, and that of HighRes3DNet is the opposite. This is because DeepMedic has a small receptive field and tends to rely on local features. It is difficult to use local features to deal with T1c, due to its complex appearance but easier to deal with FLAIR since the appearance is less complex. HighRes3DNet has a more complex model and tends to over-fit the tumor core. In contrast, PC-Net achieves a more stable performance on tumor cores and previously unseen whole tumors. The average machine time for 3D GrabCut, DeepMedic, and PC-Net is 3.87s, 65.31s and 3.83s, respectively (on the laptop), and that for HighRes3DNet is 1.10s (on the cluster).

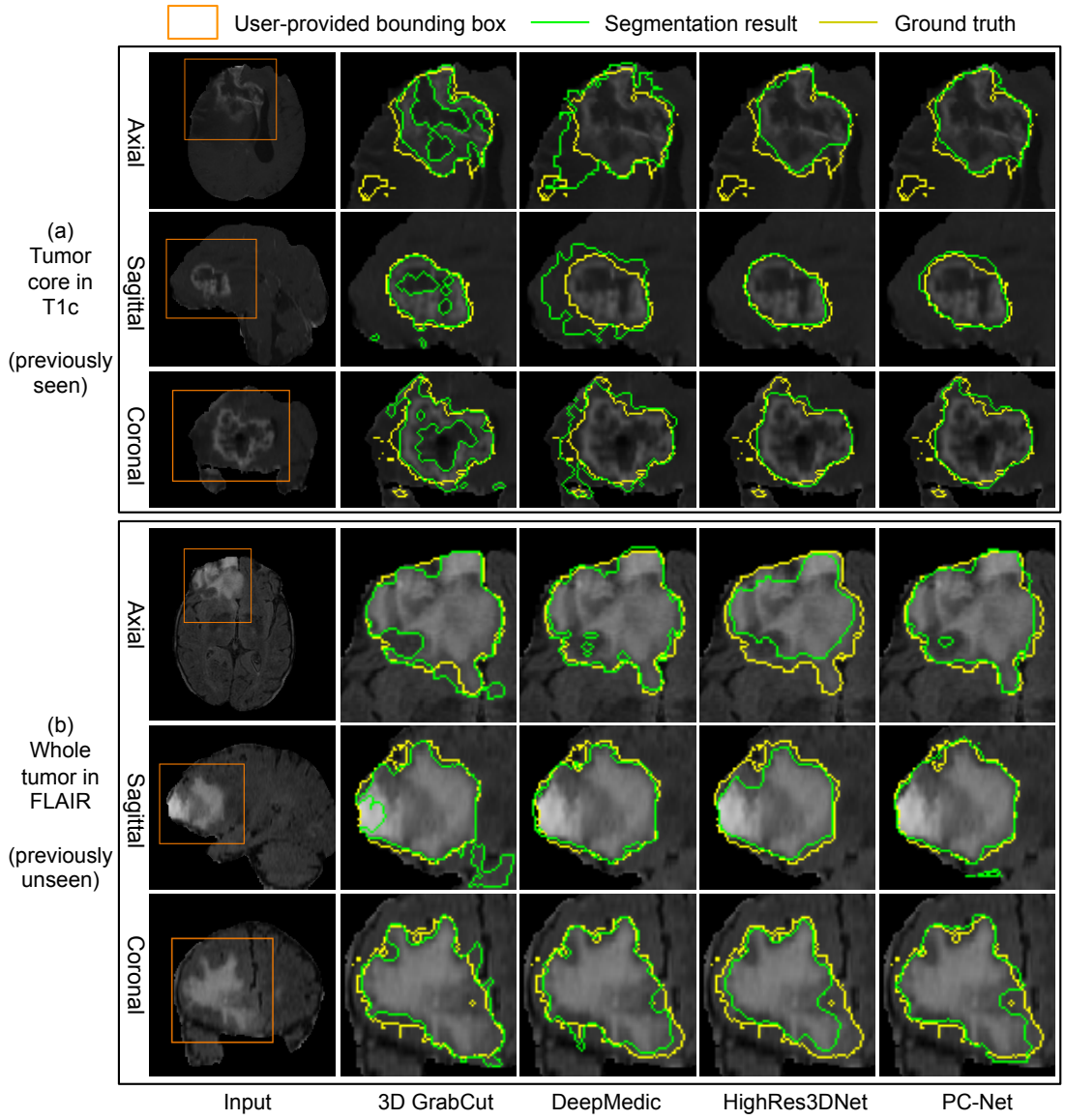


Figure 6.8: Visual comparison of initial segmentation of 3D brain tumors from a bounding box. GrabCut learns from a single image. The other methods are trained on T1c with tumor cores. The whole tumor in FLAIR is previously unseen in the training set. All the methods use the same bounding box for each test image.

6.3.3.3 Unsupervised Image-specific Fine-tuning

Fig. 6.9 shows unsupervised fine-tuning for brain tumor segmentation based on the initial output of PC-Net without additional user interactions. In Fig. 6.9(a), the tumor core is under-segmented in the initial output of PC-Net. CRF improves the segmentation to some degree, but large areas of under-segmentation still exist. The segmentation result of BIFSeg(-w) is similar to that of CRF. In contrast, BIFSeg performs better than CRF

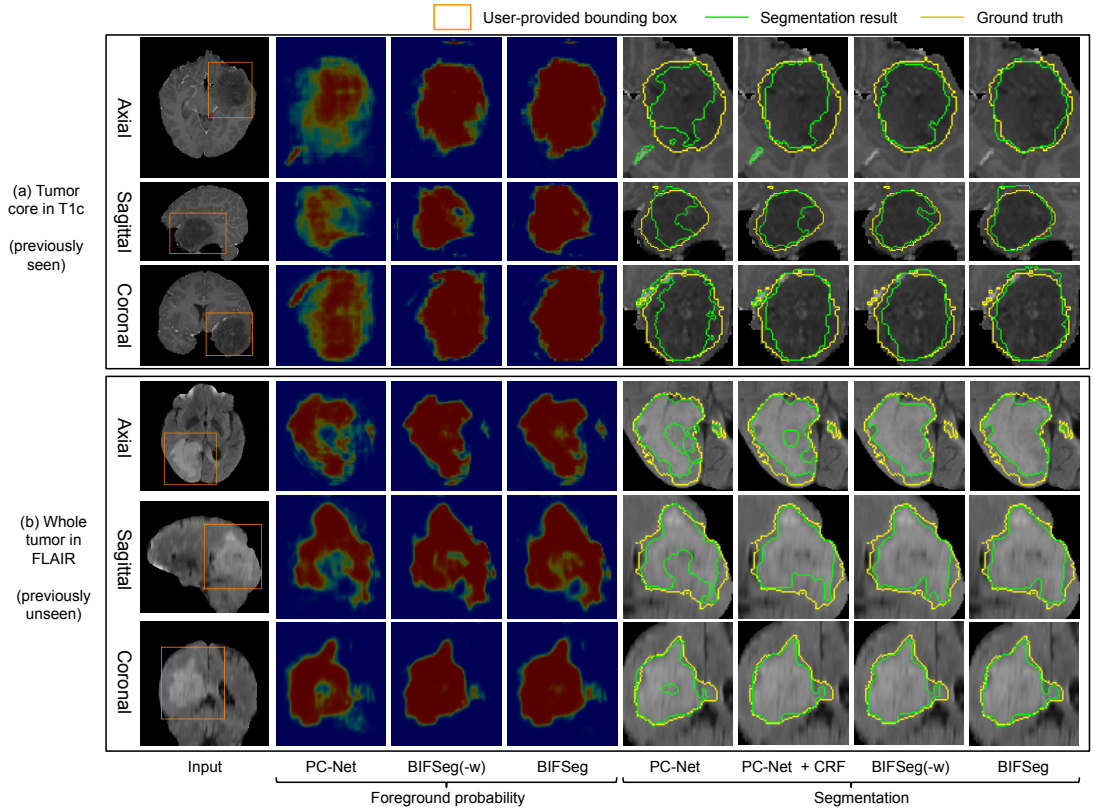


Figure 6.9: Visual comparison of PC-Net and unsupervised refinement methods without additional scribbles for 3D brain tumor segmentation. The same initial segmentation obtained by PC-Net is used by different refinement methods.

Table 6.5: Quantitative comparison of PC-Net and unsupervised refinement methods without additional scribbles for 3D brain tumor segmentation. T_m is the machine time for refinement. $\wedge$ denotes previously unseen objects. In each row, bold font denotes the best value. * denotes p-value < 0.05 compared with the others.

		PC-Net	PC-Net+CRF	BIFSeg(-w)	BIFSeg
Dice (%)	Tumor core	82.66±7.78	84.33±7.32	84.67±7.44	**86.13±6.86***
	Whole tumor $^\wedge$	83.52±8.76	83.92±7.33	83.88±8.62	**86.29±7.31***
T_m(s)	Tumor core	-	**0.12±0.04***	3.36±0.82	3.32±0.82
	Whole tumor$^\wedge$	-	**0.11±0.05***	3.16±0.89	3.09±0.83

and BIFSeg(-w). A similar situation is observed in Fig. 6.9(b) for the segmentation of previously unseen whole tumor. A quantitative comparison of these methods is shown in Table 6.5. BIFSeg improves the average Dice score from 82.66% to 86.13% for tumor core, and from 83.52% to 86.29% for whole tumor. For BIFSeg, the time to get an initial segmentation by PC-Net is less than 4s in average and the additional time for unsupervised image-specific fine-tuning is around 3s in average (Table 6.5).

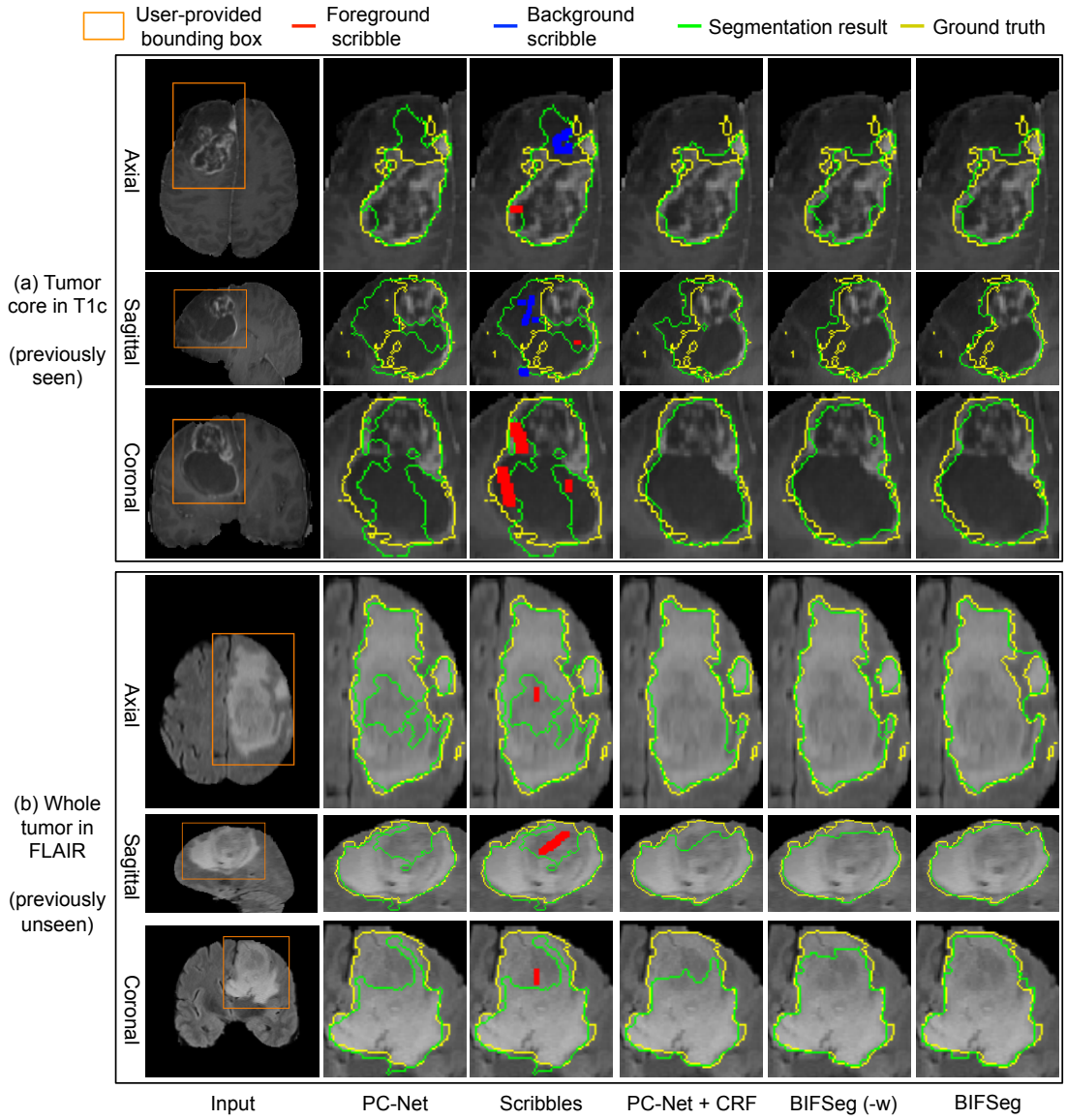


Figure 6.10: Visual comparison of PC-Net and three supervised refinement methods with scribbles for 3D brain tumor segmentation. The refinement methods use the same initial segmentation and set of scribbles.

6.3.3.4 Supervised Image-specific Fine-tuning

Fig 6.10 shows refined results of brain tumor segmentation with additional scribbles provided by the user. The same initial segmentation based on PC-Net and the same scribbles are used by CRF, BIFSeg(-w) and BIFSeg. It can be observed that CRF and BIFSeg(-w) correct the initial segmentation moderately. In contrast, BIFSeg achieves better refined results for both tumor cores in T1c and whole tumors in FLAIR. For a quantitative comparison of these refinement methods, the segmentation accuracy was

Table 6.6: Quantitative comparison of PC-Net and three supervised refinement methods with additional scribbles for 3D brain tumor segmentation. T_m is the machine time for refinement. $\wedge$ denotes previously unseen objects. In each row, bold font denotes the best value. * denotes p-value < 0.05 compared with the others.

		PC-Net	PC-Net+CRF	BIFSeg(-w)	BIFSeg
Dice (%)	Tumor core	82.66±7.78	85.93±6.64	85.88±7.53	**87.49±6.36***
	Whole tumor$^\wedge$	83.52±8.76	85.18±6.78	86.54±7.49	**88.11±6.09***
T_m(s)	Tumor core	-	**0.14±0.06***	3.33±0.86	4.42±1.88
	Whole tumor$^\wedge$	-	**0.12±0.05***	3.17±0.87	4.01±1.59

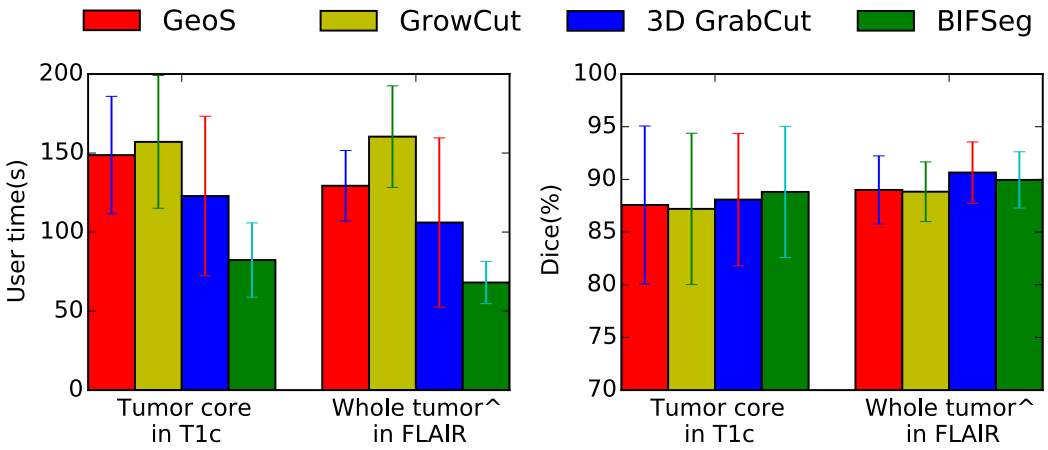


Figure 6.11: User time and Dice score of different interactive methods for 3D brain tumor segmentation. $^\wedge$ denotes previously unseen objects for BIFSeg.

measured after a single round of refinement using the same set of scribbles based on the same initial segmentation. The results are shown in Table 6.6. BIFSeg achieves an average Dice score of 87.49% and 88.11% for tumor cores and previously unseen whole tumors, respectively, and it significantly outperforms CRF and BIFSeg(-w). For supervised fine-tuning, the computational time is around 4s for one image. It is 1s longer than unsupervised fine-tuning due to the geodesic distance transform based on scribbles.

6.3.3.5 Comparison with other interactive methods

The two users (an Obstetrician and a Radiologist) used GeoS [140], GrowCut [141], 3D GrabCut [228] and BIFSeg for the brain tumor segmentation tasks respectively. For each method, the user gave interactions until the result was visually accepted. The user time and final accuracy of these methods are presented in Fig. 6.11. It shows that these interactive methods achieve similar final Dice scores for each task. However, BIFSeg

takes significantly less user time to get the results, which is 82.3s and 68.0s in average for tumor core and whole tumor, respectively.

6.4 Discussion and Conclusion

For 2D images, the proposed P-Net is trained with the placenta and the fetal brain only, but it performs well on previously unseen fetal lungs and maternal kidneys. For 3D images, the PC-Net is only trained with tumor cores in T1c, but it also achieves good results for whole tumors in FLAIR that are not present for training. This is a major advantage compared with traditional CNNs and even transfer learning [231] or weakly supervised learning [155], since for some objects it does not require annotated instances for training at all. It therefore reduces the efforts needed for gathering and annotating training data and can be applied to some unseen organs directly.

The proposed framework accepts bounding boxes and optional scribbles as user interactions. Bounding boxes in the test images are provided by the user, but they could potentially be obtained by automatic detection [190] to further increase efficiency. Compared with DeepIGeoS proposed in Chapter 5, BIFSeg does not obtain the initial segmentation automatically and requires a user-provided bounding box to start with. Drawing a bounding box can take some additional user time (about 4s for a 2D object and about 12s for a 3D object), but it provides flexibility to deal with different organs including those are not present in the training set. In addition, cropping the image with a bounding box can lead to less memory requirements and speed up the computation compared with using an entire image.

Experimental results show that the image-specific fine-tuning improves the segmentation performance. This acts as a post-processing step after the initial segmentation and outperforms CRF. Table 6.2 and 6.3 show that BIFSeg significantly outperforms CRF and BIFSeg(-w) except for the fetal brain. This is because it is relatively easy to segment the fetal brain due to its good contrast and strong edge information, so that CRF and BIFSeg(-w) can also achieve a very good performance. In contrast, segmentation of the placenta, fetal lungs and maternal kidneys is more challenging, and BIFSeg is more advantageous to deal with these organs than CRF and BIFSeg(-w).

Experiments also show that taking advantage of uncertainty plays an important role for the image-specific fine-tuning process. The uncertainty is defined based on the softmax probability and the geodesic distance to scribbles, if scribbles are given. Recent works [232] suggest that test-time dropout also provides classification uncertainty. However, test-time dropout is less suitable for interactive segmentation since it leads to longer computational time.

In conclusion, this chapter proposes an efficient deep learning-based framework for interactive 2D/3D medical image segmentation. It uses a bounding box-based CNN for binary segmentation and can segment previously unseen objects. A unified framework is proposed for both unsupervised and supervised refinements of the initial segmentation, where image-specific fine-tuning based on a weighted loss function is proposed. Experiments on segmenting multiple organs from 2D fetal MR images and brain tumors from 3D MR images show that the proposed method performs well on previously unseen objects, and the image-specific fine-tuning outperforms CRF. BIFSeg achieves similar or higher accuracy with fewer user interactions in less time than traditional interactive segmentation methods.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

This thesis presents the development of minimally interactive segmentation algorithms and their application to placenta segmentation from fetal MR images. Segmentation of the placenta is important for characterization of the placenta and fetal surgical planing. However, automatic segmentation of the placenta from fetal MR images is challenging since fetal MR images are often acquired with high 2D resolution but low 3D resolution, inter-slice motion and large inter-slice spacing. In addition, the placenta has complex variations of shape and position among patients. To address these problems, I investigated interactive segmentation of the placenta from a single 2D slice, a stack of motion-corrupted slices and multiple volumes, respectively. I used machine learning methods including Random Forests and Convolutional Neural Networks to better exploit user interactions which can lead to good segmentation results with only a few user interactions and a short user time. The developed algorithms can reduce burden on the user and provide accurate placenta segmentation efficiently.

Chapter 3 presented an ORF-based interactive method for placenta segmentation from a 2D slice. This method uses ORFs to learn from user-provided scribbles and predict the labels of the remaining pixels. When the user draws scribbles, the foreground and background scribbles are usually imbalanced, and the imbalance ratio can change during the interactive segmentation process. Traditional ORFs have a limited ability to deal with imbalanced training data with a changing imbalance ratio. I pro-

posed a generic Dynamically Balanced Online Random Forest to address this problem. I validated the proposed method through placenta segmentation from fetal MR slices and adult lung segmentation from radiographs. Experimental results showed that the proposed method achieved accurate segmentation by efficiently learning from scribbles, and demonstrated that the proposed DyBa ORF was more suitable for interactive segmentation than traditional ORFs.

Chapter 4 investigated segmentation of the placenta from a single volume (i.e., a stack of motion-corrupted slices) and multiple volumes with a minimal number of user interactions. For single volume segmentation, I proposed an efficient framework named Slic-Seg that is based on ORFs and slice-by-slice propagation. It only requires user interactions in one slice to start the segmentation process and deals with the remaining slices automatically without additional user interactions. To take advantage of complementary information of multiple volumes of the same patient, I proposed a 4D Graph Cuts-based framework to co-segment multiple volumes simultaneously using the results of single volume Slic-Seg as initialization. Experimental results showed that the single volume Slic-Seg achieved accurate results with a stable performance between and within users, and demonstrated that the co-segmentation was able to further improve the segmentation accuracy.

Chapter 5 and Chapter 6 investigated the application of deep learning to interactive image segmentation. In chapter 5, I proposed a deep interactive segmentation method (DeepIGeoS) based on CNNs and geodesic distance transforms of user interactions. DeepIGeoS uses a P-Net to obtain an initial automatic segmentation, and then uses an R-Net that takes as input the initial segmentation and user interactions to get a refined segmentation. User interactions are transformed into geodesic distance maps and used as two additional channels of the input for R-Net. I also proposed a resolution preserving network structure to avoid the potential loss of details of feature maps, and introduced a back-propagatable CRF-Net that can learn freeform pairwise potentials and leverage user interactions as hard constraints. Experimental results showed that DeepIGeoS achieved a large improvement from automatic CNNs, and obtained similar accuracy with fewer user interactions and less user time compared with traditional

interactive image segmentation methods.

In Chapter 6, I focused on the segmentation of multiple organs where only a subset of organs were annotated for training. I proposed a CNN and bounding box-based segmentation method (BIFSeg) that can deal with unseen objects. To make a pre-trained CNN to be more adaptive to a test image, I proposed unsupervised and supervised image-specific fine-tuning based on a weighted loss function that took the network- and interaction-based uncertainty into consideration. Experimental results with both 2D and 3D segmentation tasks showed that BIFSeg worked well on previously unseen objects, and the fine-tuning method outperformed traditional CRFs for post-processing. BIFSeg reduces the requirement of annotated images for training, and makes the CNN-based framework more flexible to deal with different modalities and different organs.

7.2 Future Work

The works in this thesis can be extended in four aspects in the future: segmentation with unsupervised and weakly supervised learning, fetal MR image segmentation using 3D CNNs, multi-organ and multi-modal segmentation and some clinical applications of the methods proposed in this thesis.

7.2.1 Segmentation with Unsupervised and Weakly Supervised Learning

In Chapter 3 and Chapter 4, the ORF-based segmentation methods learned from user interactions in a test image. They are flexible to deal with complex variations of appearance, shape and position of the placenta among different images. However, they used manually designed features for the learning of ORFs. These features were selected based on experience and calculated from local patches, therefore this may limit the performance of these ORF-based segmentation methods. Automatic unsupervised feature learning has shown to be more effective in several studies [233]. For example, the study in [234] has shown that data-adaptive features obtained by unsupervised learning with stacked auto-encoders outperformed hand-crafted features for MR brain image registration. In [235], unsupervised feature learning was used for multiple or-

gan detection in MR images. Such methods can learn high-level features automatically without annotations. Therefore, in the future, they can be used to learn features from an unlabeled dataset during offline training. These learned features can be used to train an ORF in an online fashion during the interactive segmentation.

In Chapter 5 and Chapter 6, the CNNs were trained with a large number of images and they required pixel-level annotations. Giving pixel-level annotations for a large dataset is very time-consuming and difficult. Thus, reducing the requirement of annotations for training is highly desired. Though BIFSeg investigated the problem of dealing with unseen objects for which annotations are not provided, it still needed full annotations of a subset of organs for training. Recently, several weakly supervised CNNs have been proposed for image segmentation. For example, ScribleSup [154] trains CNNs for semantic segmentation supervised by scribbles instead of full annotations. DeepCut [155] uses bounding boxes as annotations to train CNNs for segmentation of fetal brain and fetal lungs. These works show that it is promising to use weakly supervised learning for placenta segmentation.

7.2.2 Fetal MR Image Segmentation Using 3D CNNs

Chapter 5 and Chapter 6 validated the proposed DeepIGeoS and BIFSeg with 2D fetal MR slices. These methods can be extended to deal with 3D segmentation of the placenta. Differently from the slice-by-slice propagation in Slic-Seg, 3D CNNs allow an end-to-end prediction for volumetric images by encoding the 3D information in the networks. In this context, some specific designs of the networks may be necessary. For example, dealing with the motion between slices may need to be considered for 3D convolution. In addition, stacks of fetal MR slices have an anisotropic resolution due to the large inter-slice spacing. Therefore, convolution with anisotropic kernels can be more suitable for such images. I have proposed the idea of anisotropic convolution for multi-modal brain tumor segmentation in BraTS challenge 2017 [225]. That work uses anisotropic networks that take a stack of slices as input with a large receptive field in 2D and a relatively small receptive field in the out-plane direction that is orthogonal to the 2D slices. The anisotropic networks achieved good performance for brain tumor segmentation, and they are also suitable for fetal MR images with anisotropic

resolution.

The development of image reconstruction techniques can improve the quality of motion-corrupted images. For example, in [236], an efficient total variation algorithm was proposed for fetal brain reconstruction. In [**?**], a patch-to-volume registration approach was proposed to reconstruct fetal MR images with a large field of view. Recently, a point-spread-function-aware slice-to-volume registration approach was proposed for abdominal MR image reconstruction [237] and it can be potentially used for fetal MR image reconstruction. In the future, it is of interest to segment the placenta from a reconstructed fetal MR volume with a high 3D resolution.

7.2.3 Multi-organ and Multi-modal Segmentation

The developed algorithms in this thesis focused on the placenta, and they were proposed for binary segmentation. In clinical practice, it is often desirable to segment multiple organs for better assessment or surgical planning. For example, segmenting the fetal brain, the fetal lungs, the fetal liver, the fetal heart and the placenta can help to make a more comprehensive assessment of fetal growth. In the laser ablation therapy of TTTS, segmenting the whole fetus in addition to the placenta may help a better surgical planning. Chapter 6 has investigated the segmentation of four different organs from fetal MR images with BIFSeg. BIFSeg allows a sequential segmentation of multiple organs, but it does not support a simultaneous segmentation. Extending the developed algorithms to deal with multi-organ segmentation with higher efficiency would be of highly clinical relevance. In addition, in many applications such as brain tumor segmentation, multi-modal images are used. In Chapter 6 and Appendix B, I only took advantage of a single modality for the interactive segmentation. In [225], I investigated automatic multi-modal segmentation of brain tumors and the results show that user interactions are desirable for better robustness. Therefore, extending these proposed methods for multi-modal interactive segmentation is also of interest. The challenge may come from computational complexity, as dealing with multi-modal 3D images makes it more difficult for efficient interactive segmentation.

7.2.4 Clinical Applications

There are several potential clinical applications of the proposed segmentation methods in this thesis. First, the segmentation results of the placenta can be used for fast characterization of the placenta during pregnancy, providing detailed information about the shape, position and orientation. For example, Salafia et al. [238] investigated the effect of placenta shape on placental functional efficiency. The segmentation results lay a foundation for comprehensive shape analysis of the placenta, which can be used to estimate placental efficiency. This also helps to predict fetal growth restriction and postnatal outcome, as demonstrated in [60]. Second, a 3D segmentation result gives a more reliable measurement of the volume of the placenta compared with analyzing 2D images. This can be used to measure the growth of the placenta throughout gestation. Plasencia et al. [12] showed that the placenta volume at 11-13 weeks of gestation can be used for prediction of birth weight. Therefore, the developed segmentation methods can facilitate the birth weight prediction. Third, the segmentation results can be used as masks for high-resolution image reconstruction [68, 236] and modeling of the placenta for surgical planning and guidance [9, 61, 62]. In addition, the developed methods in Chapter 5 and Chapter 6 are also suitable for brain tumor segmentation, and they can be applied to brain tumor growth measurement and treatment planning [229].

Appendix A

Clavicle Segmentation from Chest Radiographs using DeepIGeoS

This appendix provides supplementary experimental results of applying DeepIGeoS proposed in Chapter 5 to clavicle segmentation from chest radiographs.

A.1 Clinical Background and Experimental Data

Chest radiographs are widely used for the detection and diagnosis of lung diseases such as lung cancer. Some findings on chest radiographs such as sharply circumscribed nodules or masses can indicate the presence of lung cancer. However, due to superimposition of multiple structures including ribs and clavicles, lung nodule detection and analysis is challenging. Segmenting the bone structures from chest radiographs can help to digitally suppress bones thus increase the visibility of nodules [239]. In particular, clavicle suppression might aid radiologists in detecting pathologies in the lung apex for certain lung diseases such as tuberculosis. Thus, accurate clavicle segmentation is needed to improve pathology detection. This task is challenging due to low contrast and inhomogeneous appearance in the clavicle region resulting from superimposition of several structures. In [240], it was shown that segmenting the clavicle is more difficult than segmenting the heart and the lungs. In [241], pixel classification was combined with an active shape model for automatic clavicle segmentation, while the result showed large mis-segmented areas in some images.

Table A.1: Quantitative comparison of clavicle segmentation by different networks and CRFs. Significant improvement from P-Net (p-value < 0.05) is shown in bold font.

Method	Dice(%)	ASSD(pixels)
FCN [123]	81.08±13.73	3.38±2.31
DeepLab [125]	82.27±10.80	3.09±1.47
P-Net(b5)	82.15±11.08	3.21±1.91
P-Net	84.18±10.94	2.79±1.78
P-Net + Dense CRF	83.52±11.69	2.84±1.86
P-Net + CRF-Net(g)	84.51±10.45	2.72±1.57
P-Net + CRF-Net(f)	**84.83±10.52**	**2.65±1.52**

Table A.2: Quantitative comparison of different refinement methods for clavicle segmentation. The initial segmentation is automatically obtained by P-Net + CRF-Net(f). R-Net(Euc) uses Euclidean distance instead of geodesic distance. Significant improvement from R-Net (p-value < 0.05) is shown in bold font.

Method	Dice(%)	ASSD(pixels)
Before refinement	84.83±10.52	2.65±1.52
Min-cut user-editing	87.45±8.73	2.29±1.34
R-Net(Euc)	88.34±8.91	2.20±1.17
R-Net	89.33±7.85	1.86±1.02
R-Net(Euc) + CRF-Net(fu)	88.83±8.32	1.96±1.09
R-Net + CRF-Net(fu)	**90.22±6.41**	**1.73±0.87**

This experiment uses the publicly available JSRT database[1] which consists of 247 radiographs with image resolution 2048×2048 and pixel size 0.175 mm×0.175 mm. Ground truth of 93 images were provided by the SCR database[2] based on manual segmentation by an expert. The ground truth delineated the part of clavicle projected over the lungs and mediastinum. Data in the SCR database had been split into two groups with 47 and 46 images respectively. For the first group, this experiment used 40 images as training data and the other 7 images as validation data. All the images in the second group were used as testing data. Each original image was downsampled into a size of 512×512 pixels and manually cropped with two 200×160 boxes covering the left and right clavicles respectively. The DeepIGeoS method proposed in Chapter 5 was employed for experiments. The implementation details have been described in Section 5.2.4. The following results are presented in the same way as Section 5.3.

[1] `http://www.jsrt.or.jp/jsrt-db/eng.php`
[2] `http://www.isi.uu.nl/Research/Databases/SCR`

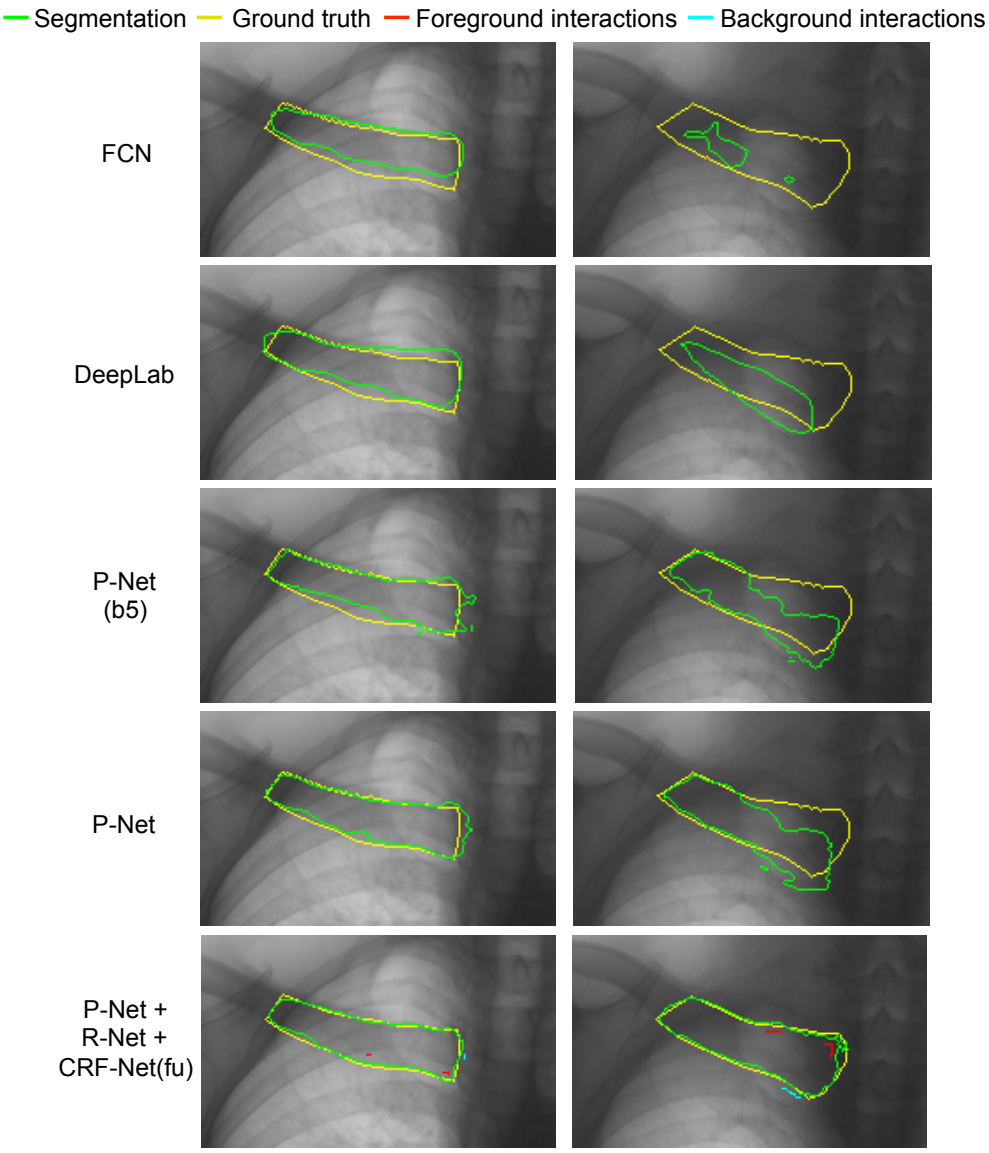


Figure A.1: Initial automatic segmentation results of the clavicle by different networks. The last row shows interactively refined results by DeepIGeoS.

A.2 Results

A.2.1 Automatic Segmentation by P-Net with CRF-Net(f)

Fig. A.1 shows examples of automatic segmentation of the clavicle by P-Net, which is compared with FCN [123], DeepLab [125] and P-Net(b5). In the first case, FCN segments the clavicle roughly, with some missed regions near the boundary. DeepLab reduces the missed regions but leads to some over-segmentation. P-Net(b5) obtains a result similar to that of DeepLab. In contrast, P-Net achieves a more accurate seg-

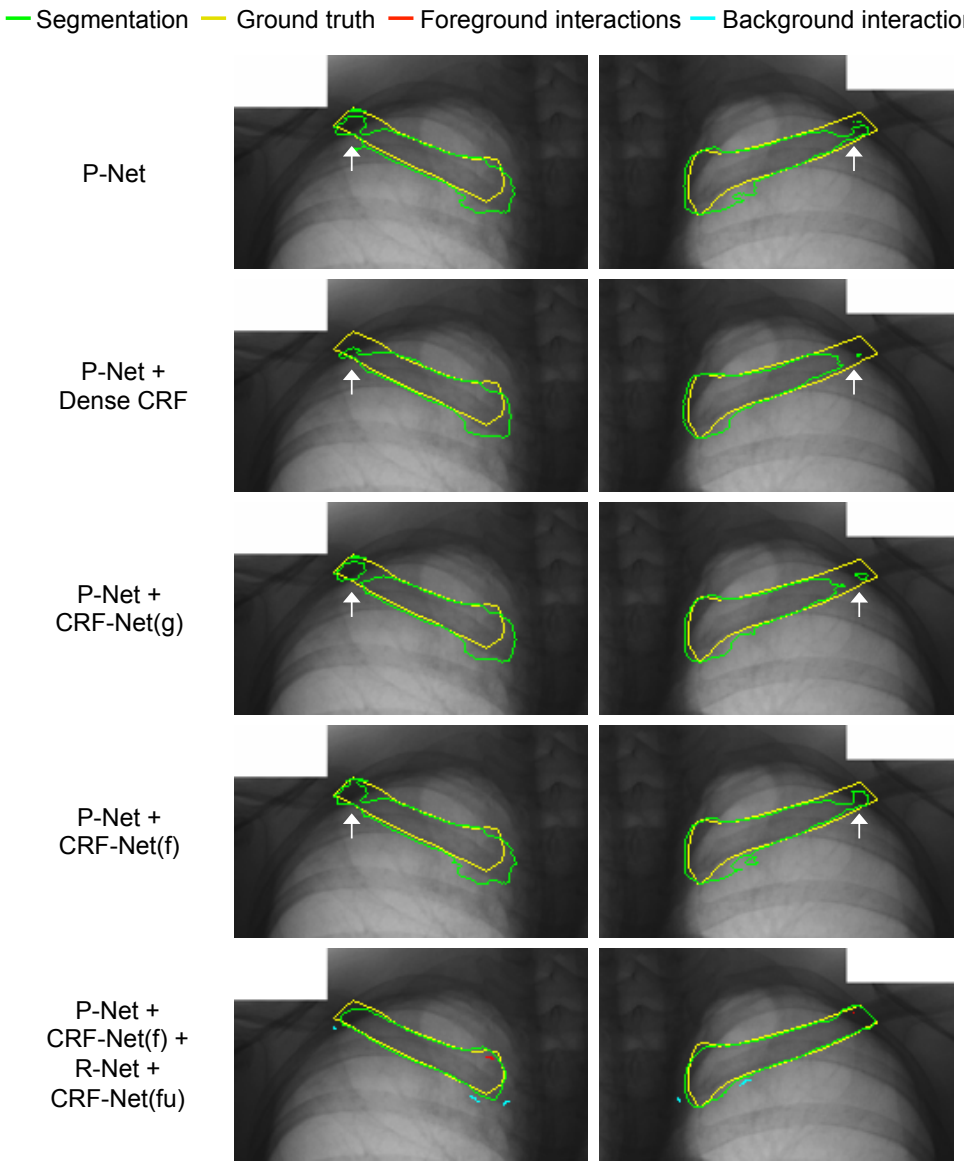


Figure A.2: Visual comparison of clavicle segmentation by P-Net with different CRFs. The last row shows interactively refined results by DeepIGeoS.

mentation which is closer to the ground truth. In the second case, FCN only captures a small region of the clavicle, while DeepLab captures a larger region with both under-segmentation and over-segmentation. P-Net(b5) and P-Net obtain better results compared with FCN and DeepLab. A quantitative evaluation of these four networks is presented in Table A.1. The result shows FCN has the lowest performance. P-Net achieves the most accurate segmentation compared with the other three networks. It achieves 84.18±10.94% in terms of Dice and 2.79±1.78 pixels in terms of ASSD.

The effect of different types of CRFs working with P-Net is shown in Fig. A.2.

It can be observed that CRF-Net(f) improves the segmentation better than Dense CRF and CRF-Net(g). A quantitative measurement of different CRFs is listed in Table A.1. The result shows only CRF-Net(f) obtains significantly better segmentation than P-Net with p-value < 0.05.

A.2.2 Interactive Refinement by R-Net with CRF-Net(fu)

Fig. A.3 shows examples of interactive refinement of clavicle segmentation using R-Net with CRF-Net(fu). The first row shows initial segmentation obtained by P-Net + CRF-Net(f). User interactions are given on that result to indicate mis-segmented areas. With the same set of user interactions, this section compares the refined results of five methods: min-cut user editing [136] and R-Net using geodesic or Euclidean distance transforms with or without CRF-Net(fu). Fig. A.3 shows that the segmentation is largely improved by refinements. The white arrows show the local difference between these five refinement methods. It can be found that more accurate results are obtained by using geodesic distance than using Euclidean distance, and CRF-Net(fu) can further help to improve the segmentation. For a quantitative comparison, I measured the segmentation accuracy after the first iteration of user refinement (applying R-Net once) using these methods with the same set of scribbles. The quantitative evaluation is listed in Table A.2, showing that the proposed R-Net with geodesic distance and CRF-Net(fu) achieves higher accuracy than the other variants, with a Dice score of 90.22$\pm$6.41% and ASSD of 1.73$\pm$0.87 pixels.

A.2.3 Comparison with Other Interactive Methods

Fig. A.4 compares DeepIGeoS with Geodesic Framework [221], Graph Cuts [133], Random Walks [222] and Slic-Seg [76] for clavicle segmentation. The first column shows initial scribbles (except for DeepIGeoS) and the resulting segmentation. The second column shows final refined results with the entire set of scribbles. The initial automatic segmentation by DeepGeoS has some errors at the head of the clavicle, and it is refined by only two short strokes given by the user. In contrast, the other four interactive methods rely on a large amount of interactions for initial segmentation, and the additional scribbles given for refinement are also long. A quantitative comparison

between these methods based on the results given by two users (an Obstetrician and a Radiologist) is shown in Fig. A.5. Compared with the traditional interactive methods, DeepIGeoS achieves similar Dice and ASSD values, but with far fewer scribbles and less user time.

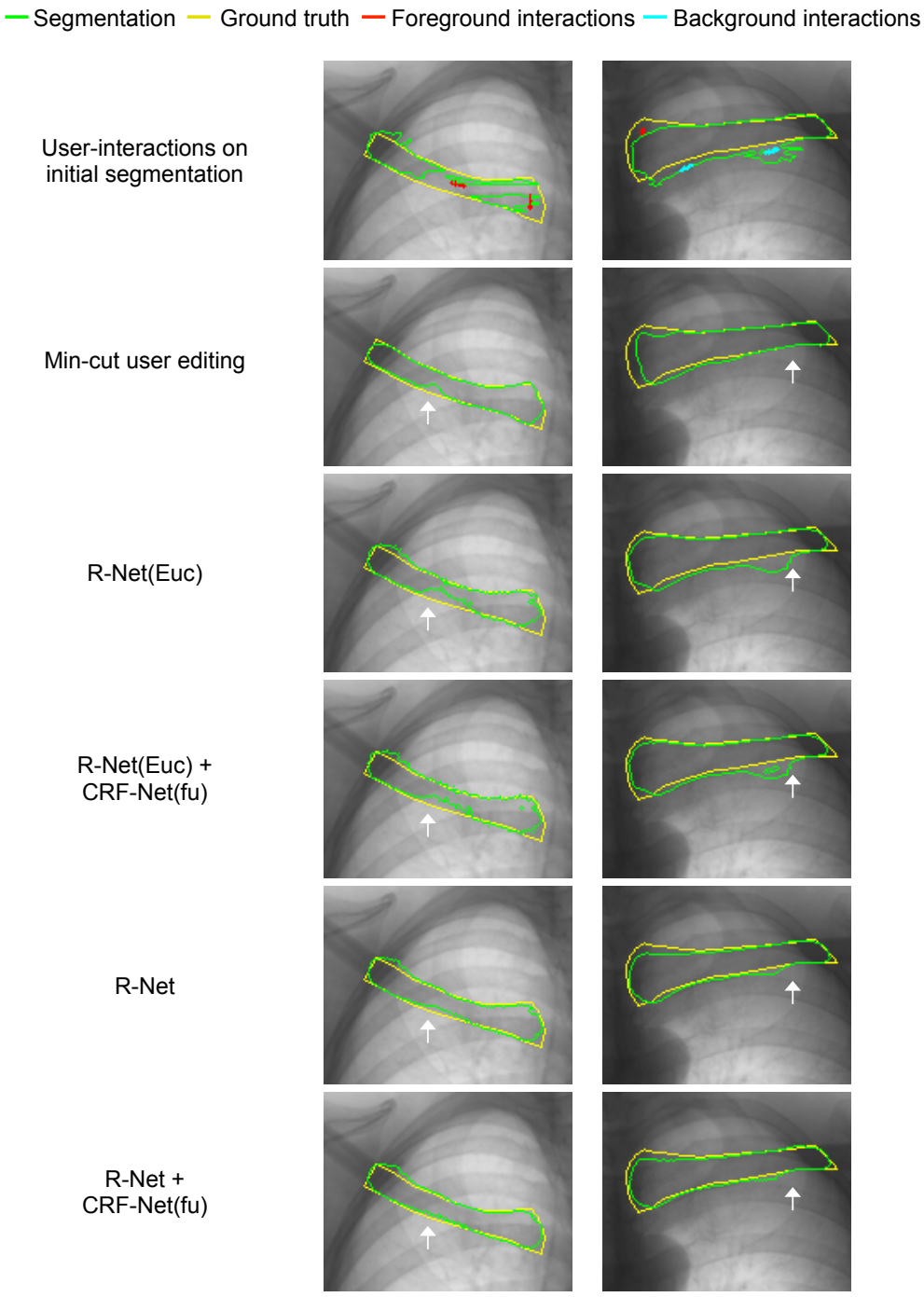


Figure A.3: Visual comparison of different refinement methods for clavicle segmentation. The first row shows the initial automatic segmentation obtained by P-Net + CRF-Net(f), on which user interactions are added for refinement. The remaining rows show refined results. R-Net(Euc) is a counterpart of the proposed R-Net and uses Euclidean distance.

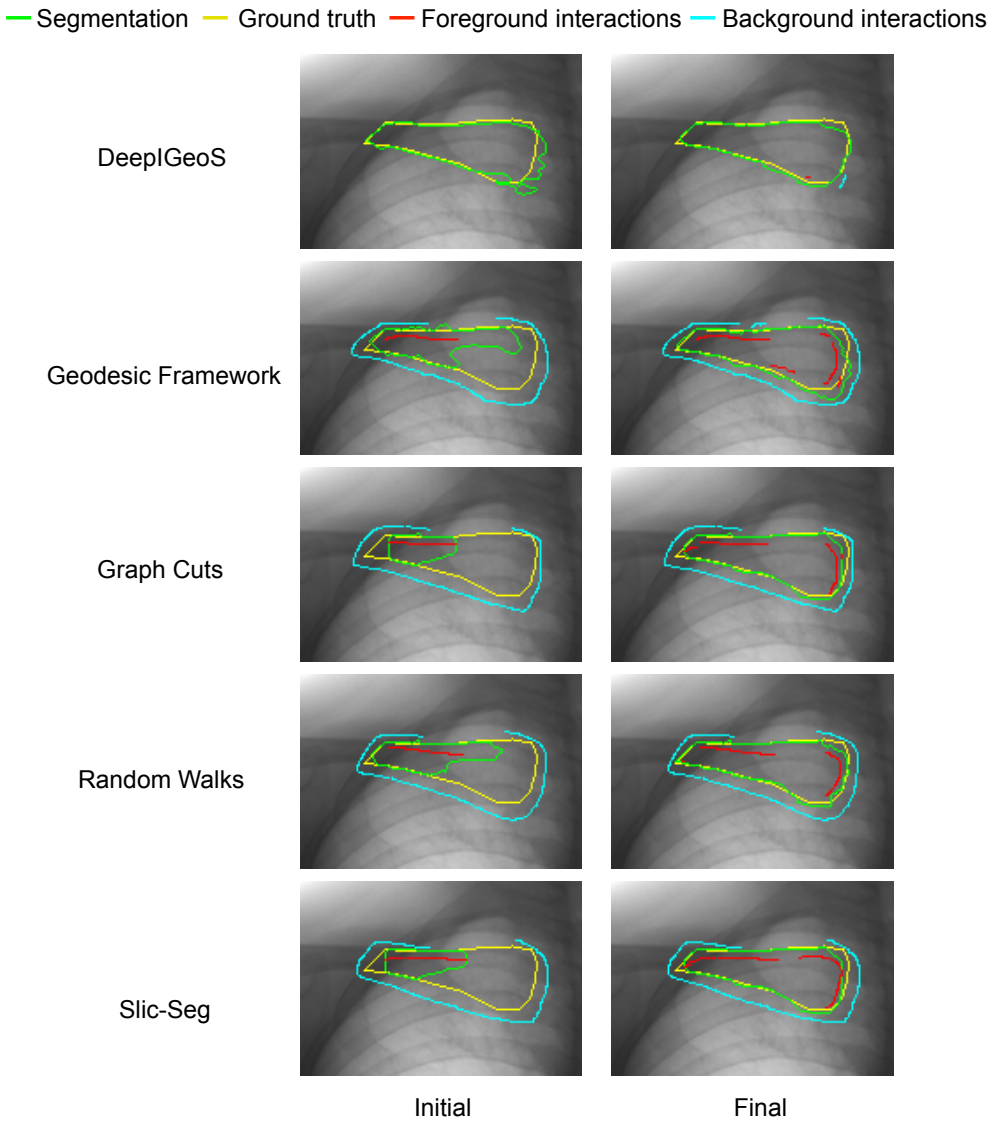


Figure A.4: Visual comparison of DeepIGeoS and other interactive methods for clavicle segmentation. The first column shows initial scribbles (except for DeepIGeoS) and the resulting segmentation. The second column shows final refined results with the entire set of scribbles. The user decided on the level of interaction required to achieve a visually acceptable result.

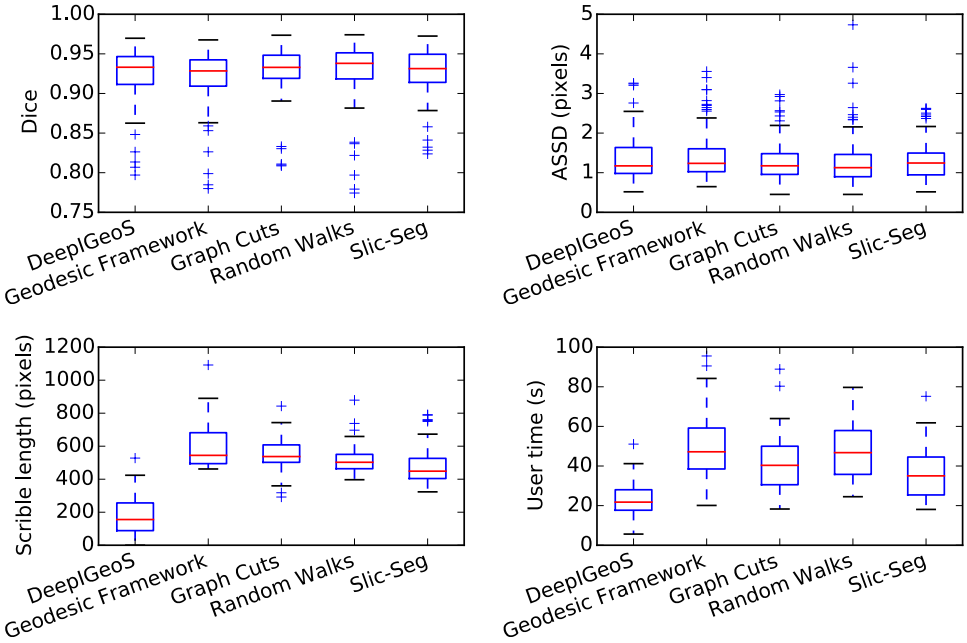


Figure A.5: Quantitative comparison of clavicle segmentation by different interactive methods in terms of Dice, ASSD, total interactions (scribble length) and user time.

Appendix B

3D DeepIGeoS for Brain Tumor Segmentation

In this appendix, I present a 3D version of DeepIGeoS and its application to 3D brain tumor segmentation. This is based on part of my article published in TPAMI [77].

B.1 Data

The clinical background of brain tumor segmentation has been introduced in Section 6.3.3. In this appendix, I investigate interactive segmentation of the whole tumor from FLAIR images. The 2015 Brain Tumor Segmentation Challenge (BraTS) [229] training set with images of 274 cases were used in this experiment. The ground truth were manually delineated by several experts. As a first demonstration of deep interactive segmentation in 3D, I only use FLAIR images in the dataset and only segment the whole tumor. I randomly selected 234 cases for training and used the remaining 40 cases for testing. All these images had been skull-stripped and resampled to size of $240\times240\times155$ with isotropic resolution 1mm^3. Each image was cropped based on the bounding box of its non-zero region.

B.2 3D Networks and Implementation

For 3D segmentation, this chapter reuses the PC-Net presented in Fig. 6.2 that is an extension of P-Net proposed in Chapter 5 (Fig. 5.3). To make it clear that this network works on 3D images, this chapter refers to it as 3D P-Net. The segmentation pro-

Table B.1: Quantitative comparison of 3D brain tumor segmentation by different networks and CRFs. Significant improvement from 3D P-Net (p-value < 0.05) is shown in bold font.

Method	Dice(%)	ASSD(pixels)
DeepMedic [128]	83.87±8.72	2.38±1.52
HighRes3DNet [131]	85.47±8.66	2.20±2.24
3D P-Net(b5)	85.36±7.34	2.21±2.13
3D P-Net	86.68±7.67	2.14±2.17
3D P-Net + Dense CRF	87.06±7.23	2.10±2.02
3D P-Net + CRF-Net(f)	**87.55±6.72**	**2.04±1.70**

cess follows the workflow shown in Fig. 5.1 and 3D P-Net is used with a CRF-Net(f) for segmentation. The refinement network is referred to as 3D R-Net which shares the same structure as 3D P-Net except its input has three additional channels and the CRF-Net(f) is replaced by CRF-Net(fu). The geodesic distance transformation, CRF-Net(f) and CRF-Net(fu) are also extended to their 3D versions. The patch size for 3D CRF-Net is set to $5\times5\times3$ for computational efficiency. The 3D networks are implemented by Tensorflow[1] [242] using NiftyNet[2] [131]. The training process was done via two 8-core E5-2623v3 Intel Haswells and two K80 NVIDIA GPUs and 128GB memory. The testing process with user interactions was performed on a MacBook Pro (OS X 10.9.5) with 16GB RAM and an Intel Core i7 CPU running at 2.5GHz and an NVIDIA GeForce GT 750M GPU. A PyQt GUI was developed for the 3D interactive segmentation task.

B.3 Results

B.3.1 Automatic Segmentation by 3D P-Net with CRF-Net(f)

Fig. B.1 shows examples of automatic segmentation of brain tumor by 3D P-Net, which is compared with DeepMedic [128], HighRes3DNet [131] and 3D P-Net(b5) that is a variant of 3D P-Net and only uses features from block 5 (Fig. 6.2) instead of concatenated multi-scale features. In the first column, DeepMedic segments the tumor roughly, with some missed regions near the boundary. HighRes3DNet reduces the missed regions but leads to some over-segmentation. 3D P-Net(b5) obtains a similar result to

[1] `https://www.tensorflow.org`
[2] `http://niftynet.io`

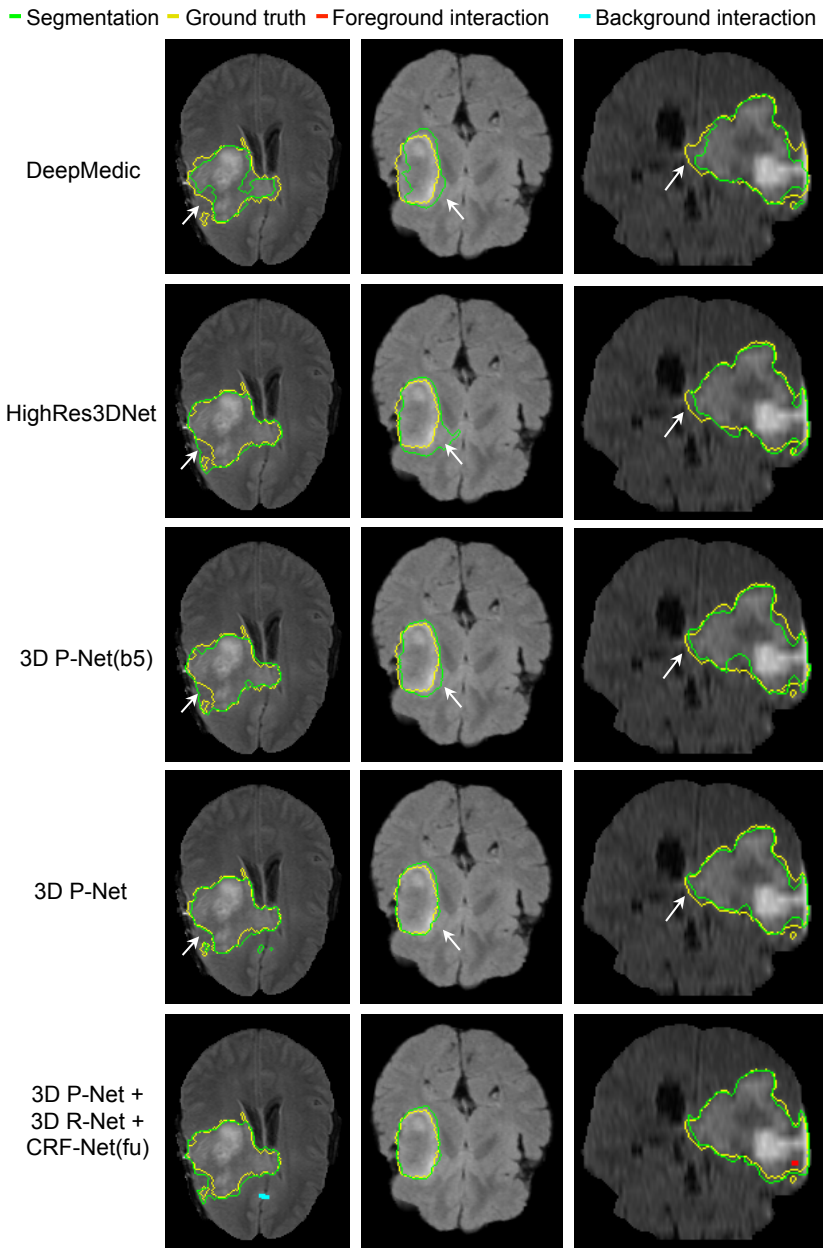


Figure B.1: Initial automatic 3D segmentation of brain tumor by different networks. The last row shows interactively refined results by DeepIGeoS.

that of HighRes3DNet. In contrast, 3D P-Net achieves a more accurate segmentation, which is closer to the ground truth. More examples in the second and third column in Fig. B.1 also show 3D P-Net outperforms the other networks. Quantitative evaluation of these four networks is presented in Table B.1. DeepMedic achieves an average dice score of 83.87%. HighRes3DNet and 3D P-Net(b5) achieve similar performance, and they are better than DeepMedic. 3D P-Net outperforms these three counterparts with

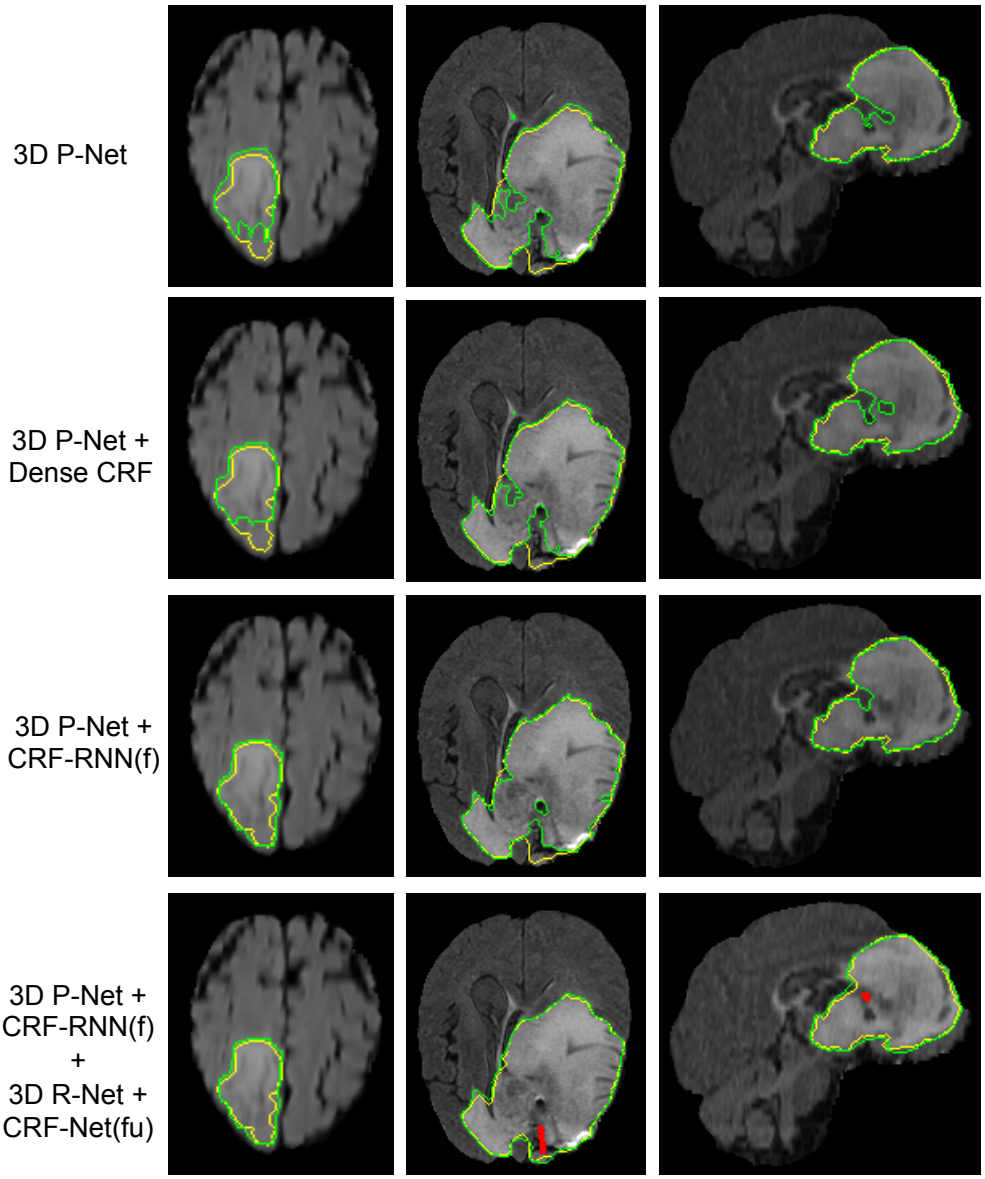


Figure B.2: Visual comparison between Dense CRF and the proposed CRF-Net(f) for 3D brain tumor segmentation. The last column shows interactively refined results by DeepIGeoS.

86.68±7.67% in terms of Dice and 2.14±2.17 pixels in terms of ASSD. Note that the proposed 3D P-Net has far fewer parameters compared with HighRes3DNet. It is more memory efficient and therefore can perform inference on a 3D volume in interactive time.

Since CRF-RNN [171] was only implemented for 2D, in the context of 3D segmentation this chapter only compared 3D CRF-Net(f) with 3D Dense CRF [128] that uses manually tuned parameters. Visual comparison between these two types of CRFs

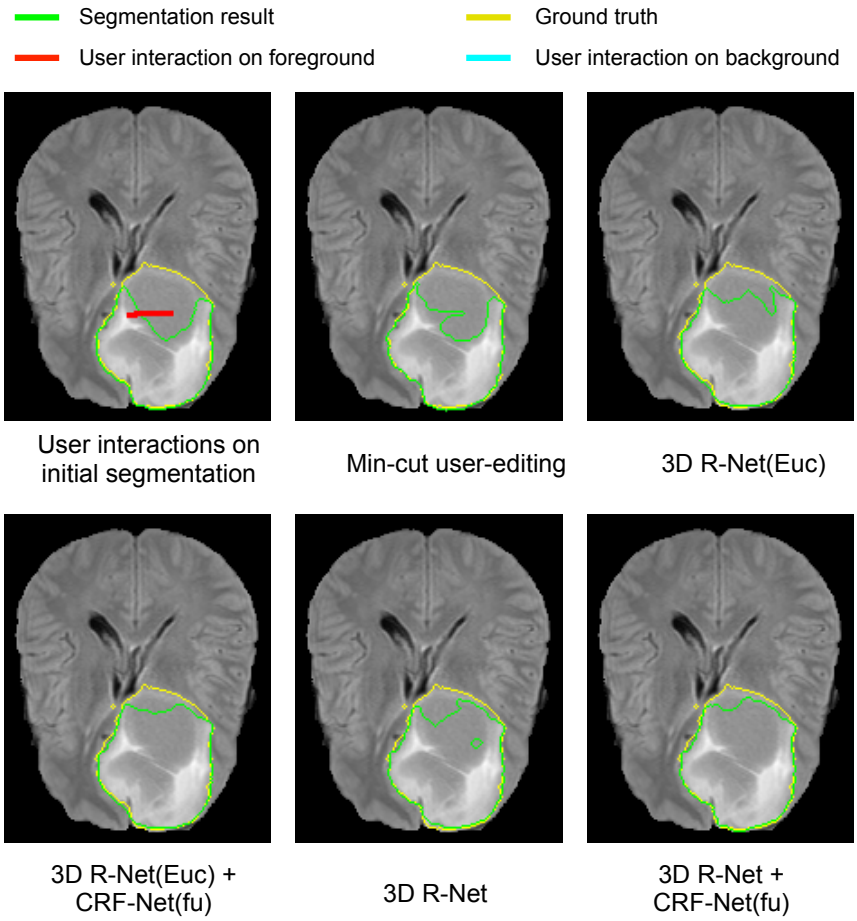


Figure B.3: Visual comparison of different refinement methods for 3D brain tumor segmentation. The initial segmentation is obtained by 3D P-Net + CRF-Net(f), on which user interactions are given. 3D R-Net(Euc) is a counterpart of the proposed 3D R-Net and it uses Euclidean distance.

working with 3D P-Net is shown in Fig. B.2. It can be observed that CRF-Net(f) achieves more noticeable improvement compared with Dense CRF that is used as post-processing without end-to-end learning. Quantitative measurement of Dense CRF and CRF-Net(f) is listed in Table B.1. It shows that only CRF-Net(f) obtains significantly better segmentation than 3D P-Net with p-value < 0.05.

B.3.2 Interactive Refinement by 3D R-Net with CRF-Net(fu)

Fig. B.3 shows examples of interactive refinement of brain tumor segmentation using 3D R-Net with CRF-Net(fu). The initial segmentation is obtained by 3D P-Net + CRF-Net(f). With the same set of user interactions, I compared the refined results of min-cut user-editing and four variations of 3D R-Net: using geodesic or Euclidean

Table B.2: Quantitative comparison of different refinement methods for 3D brain tumor segmentation with the same set of scribbles. The segmentation before refinement is obtained by 3D P-Net + CRF-Net(f). 3D R-Net(Euc) uses Euclidean distance instead of geodesic distance. Significant improvement from 3D R-Net (p-value $<$ 0.05) is shown in bold font.

Method	Dice(%)	ASSD(pixels)
Before refinement	87.55±6.72	2.04±1.70
Min-cut user-editing	88.41±7.05	1.74±1.53
3D R-Net(Euc)	88.82±7.68	1.60±1.56
3D R-Net	89.30±6.82	1.52±1.37
3D R-Net(Euc) + CRF-Net(fu)	89.27±7.32	1.48±1.22
3D R-Net + CRF-Net(fu)	**89.93±6.49**	**1.43±1.16**

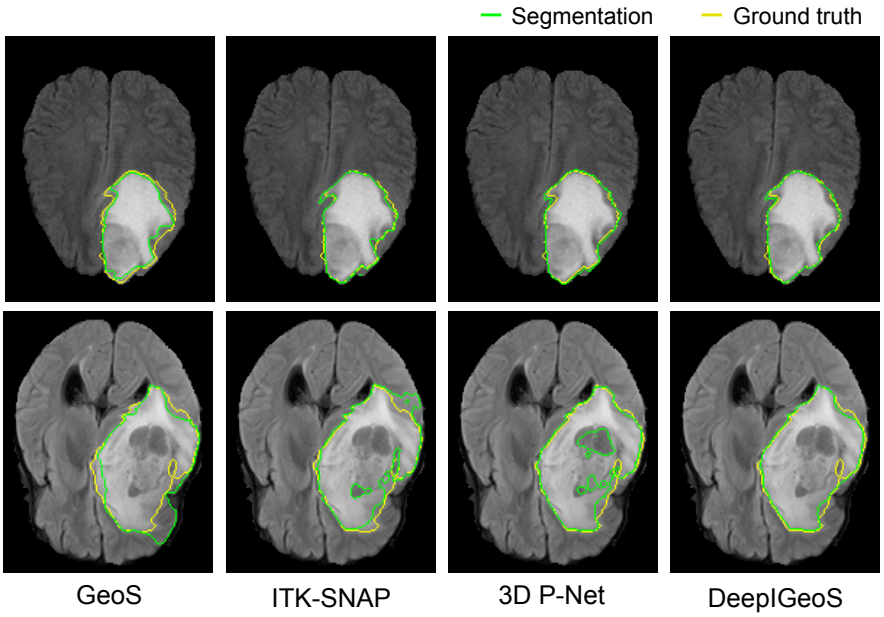


Figure B.4: Visual comparison of 3D brain tumor segmentation using GeoS, ITK-SNAP, and DeepIGeoS that is based on 3D P-Net.

distance transforms with or without CRF-Net(fu). Fig. B.3 shows that min-cut user-editing achieves a small improvement.

It can be found that more accurate results are obtained by using geodesic distance than using Euclidean distance, and CRF-Net(fu) can further help to improve the segmentation. For quantitative comparison, I measured the segmentation accuracy after the first iteration of refinement, in which the same set of scribbles were used for different refinement methods. The quantitative evaluation is listed in Table B.2, showing that the proposed 3D R-Net with geodesic distance and CRF-Net(fu) achieves higher

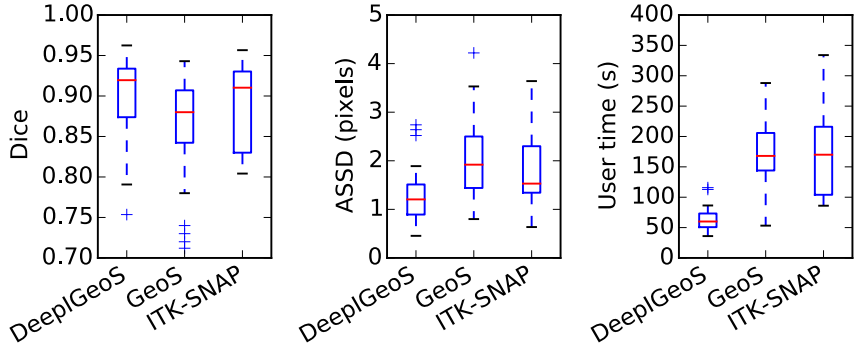


Figure B.5: Quantitative evaluation of 3D brain tumor segmentation by DeepIGeoS, GeoS and ITK-SNAP.

accuracy than the other variations with a Dice score of 89.93±6.49% and ASSD of 1.43±1.16 pixels.

B.3.3 Comparison with Other Interactive Methods

Fig. B.4 shows a visual comparison between GeoS [140], ITK-SNAP [138] and DeepIGeoS. In the first row, the tumor has a good contrast with the background. All the compared methods achieve very accurate segmentations. In the second row, a lower contrast makes it difficult for the user to identify the tumor boundary. Benefited from the initial tumor boundary that is automatically identified by 3D P-Net, DeepIGeoS outperforms GeoS and ITK-SNAP. Quantitative comparison is presented in Fig. B.5. It shows DeepIGeoS achieves higher accuracy compared with GeoS and ITK-SNAP. In addition, the user time for DeepIGeoS is about one third of that for the other two methods.

Appendix C

List of Abbreviations

AAM: Active Appearance Model

AE: Auto-Encoder

ASM: Active Shape Model

ASSD: Average Symmetric Surface Distance

b-FFE: Balanced Fast Field Echo

BRATS: Brain Tumor Segmentation Challenge

CEUS: Contrast Enhanced Ultrasound

CNN: Convolutional Neural Network

CPU: Central Processing Unit

CRF: Conditional Random Field

CT: Computed Tomography

CUDA: Compute Unified Device Architecture

DWI: Diffusion Weighted Imaging

DWT: Discrete Wavelet Transform

EM: Expectation Maximization

FCN: Fully Convolutional Network

FFD: Free-Form Deformation

FLAIR: Fluid-Attenuated Inverse Recovery

FLASH: Fast Low Angle Shot

GLCM: Gray Level Co-occurrence Matrix

GMM: Gaussian Mixture Model

GPU: Graphics Processing Unit

GUI: Graphical User Interface

GVF: Gradient Vector Flow

HASTE: Half Fourier Single Shot Turbo Spin Echo

IUFD: Intra-Uterine Fetal Demise

IUGR: Intra-Uterine Growth Restriction

MRI: Magnetic Resonance Imaging

MRS: Magnetic Resonance Spectroscopy

ORF: Online Random Forest

PET: Positron Emission Tomography

RAM: Random Access Memory

RF: Random Forest

RNN: Recurrent Neural Network

ROI: Region of Interest

SGD: Stochastic Gradient Decent

SIUGR: Selective Intra-Uterine Growth Restriction

SSFP: Steady State Free Precession

SSFSE: Single Shot Fast Spin Echo

SVM: Support Vector Machine

T1c: Contrast Enhanced T1-Weighted Imaging

TTTS: Twin-Twin Transfusion Syndrome

Bibliography

[1] Carlos Bermúdez, Carlos H. Becerra, Patricia W. Bornick, Mary H. Allen, Jorge Arroyo, and Rubén A. Quintero. Placental types and twin-twin transfusion syndrome. *American Journal of Obstetrics and Gynecology*, 187(2):489–494, 2002.

[2] Jan A. Deprest, Alan W. Flake, Eduard Gratacos, Yves Ville, Kurt Hecher, Kypros Nicolaides, Mark P. Johnson, François I. Luks, N. Scott Adzick, and Michael R. Harrison. The making of fetal surgery. *Prenatal Diagnosis*, 30(7):653–667, 2010.

[3] Tony Y. T. Tan and George S. H. Yeo. Intrauterine growth restriction. *Current opinion in obstetrics & gynecology*, 17(2):135–142, 2005.

[4] Andreea A. Creanga, Cynthia J. Berg, Carla Syverson, Kristi Seed, F. Carol Bruce, and William M. Callaghan. Pregnancy-related mortality in the United States, 2006-2010. *Obstetrics & Gynecology*, 125(1):5–12, 2015.

[5] Outi Hovatta, Arja Lipasti, Juhani Rapola, and Olavi Karjalainen. Causes of stillbirth: a clinicopathological study of 243 patients. *BJOG: An International Journal of Obstetrics & Gynaecology*, 90(8):691–696, 1983.

[6] Kevin Kearney, Nicole Vigneron, Patty Frischman, and John WC Johnson. Fetal weight estimation by ultrasonic measurement of abdominal circumference. *Obstetrics and gynecology*, 51(2):156–162, 1978.

[7] Carri R. Warshak, Ramez Eskander, Andrew D. Hull, Angela L. Scioscia, Robert F. Mattrey, Kurt Benirschke, and Robert Resnik. Accuracy of ultra-

sonography and magnetic resonance imaging in the diagnosis of placenta accreta. *Obstetrics and gynecology*, 108(3):573–81, 2006.

[8] Marie-Victoire Senat, Jan Deprest, Michel Boulvain, Alain Paupe, Norbert Winer, and Yves Ville. Endoscopic laser surgery versus serial amnioreduction for severe twin-to-twin transfusion syndrome. *The New England journal of medicine*, 351(2):136–144, 2004.

[9] Rosalind Pratt, Jan Deprest, Tom Vercauteren, Sebastien Ourselin, and Anna L. David. Computer-assisted surgical planning and intraoperative guidance in fetal surgery: a systematic review. *Prenatal Diagnosis*, 35(12):1159–1166, 2015.

[10] Biology forums [online], 2013.

[11] Manop Janthanaphan, Ounjai Kor-Anantakul, and Alan Geater. Placental weight and its ratio to birth weight in normal pregnancy at Songkhlanagarind Hospital. *Journal of the Medical Association of Thailand*, 89(2):130–137, 2006.

[12] Walter Plasencia, Ranjit Akolekar, Themistoklis Dagklis, Alina Veduta, and Kypros H. Nicolaides. Placental volume at 11-13 weeks' gestation in the prediction of birth weight percentile. *Fetal Diagnosis and Therapy*, 30:23–28, 2011.

[13] Luz Helena Sanin, Sandra Reza Lopez, Edith Tufiño Olivares, Martha Corral Terrazas, Miguel Angel Robles Silva, and Margarita Levario Carrillo. Relation between birth weight and placenta weight. *Biology of the Neonate*, 80(2):113–117, 2001.

[14] Christina Y. L. Aye, Gordon N. Stevenson, Lawrence Impey, and Sally L. Collins. Comparison of 2-D and 3-D estimates of placental volume in early pregnancy. *Ultrasound in Medicine and Biology*, 41(3):734–740, 2015.

[15] Sabaratnam Arulkumaran, Mahantesh Karoshi, Louis G. Keith, Andre B. Lalonde, and Christopher B-Lynch. Placental abnormalities. In *A comprehensive textbook of postpartum hemorrhage*, pages 80–97. 2006.

[16] Singapore general hospital [online], 2016.

[17] Nada El Garhy. Placenta previa - causes, types and management, 2016.

[18] Christine H. Comstock. The antenatal diagnosis of placental attachment disorders. *Current Opinion in Obstetrics and Gynecology*, 23(2):117–122, 2011.

[19] Abubakar A. Panti, Bissala A. Ekele, Emmanuel I. Nwobodo, and Ahmed Yakubu. The relationship between the weight of the placenta and birth weight of the neonate in a Nigerian hospital. *Nigerian Medical Journal : Journal of the Nigeria Medical Association*, 53(2):80–84, 2012.

[20] Giorgio Pardi, Anna Maria Marconi, and Irene Cetin. Placental-fetal interrelationship in IUGR fetuses - A review. *Placenta*, 23(SUPPL. 1):S136–S141, 2002.

[21] Japan fetal therapy group [online], 2017.

[22] Corinne Hubinont, Liesbeth Lewi, Pierre Bernard, Etienne Marbaix, Frédéric Debième, and Eric Jauniaux. Anomalies of the placenta and umbilical cord in twin gestations. *American Journal of Obstetrics and Gynecology*, 213(4):S91–S102, 2015.

[23] Tim Van Mieghem, Liesbeth Lewi, Léonardo Gucciardo, Philip Dekoninck, Dominique Van Schoubroeck, Roland Devlieger, and Jan Deprest. The fetal heart in twin-to-twin transfusion syndrome. *International Journal of Pediatrics*, 2010(9):1–8, 2010.

[24] Dan V. Valsky, Elisenda Eixarch, Josep Maria Martinez, Fatima Crispi, and Eduard Gratacós. Selective intrauterine growth restriction in monochorionic twins: Pathophysiology, diagnostic approach and management dilemmas. *Seminars in Fetal and Neonatal Medicine*, 15(6):342–348, 2010.

[25] Liesbeth Lewi, Leonardo Gucciardo, Agnes Huber, Jacques Jani, Tim Van Mieghem, Elisa Doné, Mieke Cannie, Eduardo Gratacós, Anke Diemert, Kurt Hecher, Paul Lewi, and Jan Deprest. Clinical outcome and placental characteristics of monochorionic diamniotic twin pairs with early- and late-onset discor-

dant growth. *American Journal of Obstetrics and Gynecology*, 199(5):511.e1 – 511.e, 2008.

[26] Cande V. Ananth, Anthony M. Vintzileos, Susan Shen-Schwarz, John C. Smulian, and Yu Ling Lai. Standards of birth weight in twin gestations stratified by placental chorionicity. *Obstetrics and Gynecology*, 91(6):917–924, 1998.

[27] Rubeén A. Quintero, Patricia W. Bornick, Walter J. Morales, and Mary H. Allen. Selective photocoagulation of communicating vessels in the treatment of monochorionic twins with selective growth retardation. *American Journal of Obstetrics and Gynecology*, 185(3):689–696, 2001.

[28] Wikipedia [online], 2015.

[29] Thomas L. Szabo. *Diagnostic ultrasound imaging: inside out*. 2014.

[30] Thomas R. Nelson and Dolores H. Pretorius. Three-dimensional ultrasound imaging. *Ultrasound in Medicine & Biology*, 24(9):1243–1270, 1998.

[31] Alec Welsh, Minsheng Hou, Neama Meriki, and Gordon Stevenson. Use of four-dimensional analysis of power Doppler perfusion indices to demonstrate cardiac cycle pulsatility in fetoplacental flow. *Ultrasound in Medicine and Biology*, 38(8):1345–1351, 2012.

[32] Rodrigo Ruano, Alexandra Benachi, Laurence Joubin, Marie-Cécile Aubry, Jean-Christophe Thalabard, Yves Dumez, and Marc Dommergues. Three-dimensional ultrasonographic assessment of fetal lung volume as prognostic factor in isolated congenital diaphragmatic hernia. *BJOG : an international journal of obstetrics and gynaecology*, 111(5):423–9, 2004.

[33] Chiung-Hsin Chang, Fong-Ming Chang, Chen-Hsiang Yu, Huei-Chen Ko, and Hsi-Yao Chen. Assessment of fetal cerebellar volume using three-dimensional ultrasound. *Ultrasound in Medicine & Biology*, 26(6):981–988, 2000.

[34] Giuseppe Rizzo, Alessandra Capponi, Ottavia Cavicchioni, Marianne Vendola, and Domenico Arduini. First trimester uterine Doppler and three-dimensional

ultrasound placental volume calculation in predicting pre-eclampsia. *European Journal of Obstetrics Gynecology and Reproductive Biology*, 138(2):147–151, 2008.

[35] Yurong Hong, Xueming Liu, Zhiyu Li, Xiufang Zhang, Meifeng Chen, and Zhiyan Luo. Real-time ultrasound elastography in the differential diagnosis of benign and malignant thyroid nodules. *Journal of Ultrasound in Medicine*, 28(7):861–867, 2009.

[36] Jonathan G. Crisp, Luis M. Lovato, and Timothy B. Jang. Compression ultrasonography of the lower extremity with portable vascular ultrasonography can accurately detect deep venous thrombosis in the emergency department. *Annals of Emergency Medicine*, 56(6):601–610, 2010.

[37] Salim Daya. Accuracy of gestational age estimation by means of fetal crown-rump length measurement. *American Journal of Obstetrics and Gynecology*, 168(3):903–908, 1993.

[38] Frank P. Hadlock, Ronald B. Harrist, Russell L. Deter, and Ssung K. Park. Fetal femur length as a predictor of menstrual age: sonographically measured. *American Journal of Roentgenology*, 138(5):875–878, 1982.

[39] Derek A. Fyfe and Charles H. Kline. Fetal echocardiographic diagnosis of congenital heart disease. *Pediatric Clinics of North America*, 37(1):45–67, 1990.

[40] Lawrence Oppenheimer, Anthony Armson, Dan Farine, Lisa Keenan-Lindsay, Valerie Morin, Tracy Pressey, Marie France Delisle, Robert Gagnon, William Robert Mundle, and John Van Aerde. Diagnosis and management of placenta previa. *Journal of Obstetrics and Gynaecology Canada*, 29(3):261–266, 2007.

[41] Denise Pugash, Peter C Brugger, Dieter Bettelheim, and Daniela Prayer. Prenatal ultrasound and fetal MRI : the comparative value of each modality in prenatal diagnosis. *European journal of radiology*, 68(2):214–226, 2008.

[42] Sahar N. Saleem. Fetal magnetic resonance imaging (MRI): a tool for a better understanding of normal and abnormal brain development. *Journal of child neurology*, 28(7):890–908, 2013.

[43] Daniela Prayer, Peter Christian Brugger, and Lucas Prayer. Fetal MRI: techniques and protocols. *Pediatric Radiology*, 34:685–693, 2004.

[44] Ali Gholipour, Judith A. Estroff, Carol E. Barnewolt, Richard L. Robertson, P. Ellen Grant, Borjan Gagoski, Simon K. Warfield, Onur Afacan, Susan A. Connolly, Jeffrey J. Neil, Adam Wolfberg, and Robert V. Mulkern. Fetal MRI: A technical update with educational aspirations. *Concepts in Magnetic Resonance Part A: Bridging Education and Research*, 43(6):237–266, 2014.

[45] Sahar N. Saleem. Fetal MRI: an approach to practice: a review. *Journal of Advanced Research*, 5(5):507–523, 2014.

[46] Bin Zhu, Bing Zhang, Ming Li, Shifu Xi, Decai Yu, and Yitao Ding. An evaluation of a superfast MRI sequence in the diagnosis of suspected acute appendicitis. *Quantitative imaging in medicine and surgery*, 2(4):280–287, 2012.

[47] Sahar N. Saleem. Feasibility of MRI of the fetal heart with balanced steady-state free precession sequence along fetal body and cardiac planes. *American Journal of Roentgenology*, 191(4):1208–1215, 2008.

[48] Hao Huang, Rong Xue, Jiangyang Zhang, Tianbo Ren, Linda J. Richards, Paul Yarowsky, Michael I. Miller, and Susumu Mori. Anatomical characterization of human fetal brain development with diffusion tensor magnetic resonance imaging. *Journal of Neuroscience*, 29(13):4263–4273, 2009.

[49] Harald Marcel Bonel, Bernhard Stolz, Lars Diedrichsen, Kathrin Frei, Bettina Saar, Boris Tutschek, Luigi Raio, Daniel Surbek, Sudesh Srivastav, Mathias Nelle, Johannes Slotboom, and Roland Wiest. Diffusion-weighted MR imaging of the placenta in fetuses with placental insufficiency. *Radiology*, 257(3):810–819, 2010.

[50] Nadine Girard, Sylviane Confort Gouny, Angèle Viola, Yann Le Fur, Patrick Viout, Kathia Chaumoitre, Claude D'Ercole, Catherine Gire, Dominique Figarella-Branger, and Patrick J Cozzone. Assessment of normal fetal brain maturation in utero by proton magnetic resonance spectroscopy. *Magnetic resonance in medicine : official journal of the Society of Magnetic Resonance in Medicine / Society of Magnetic Resonance in Medicine*, 56(4):768–775, 2006.

[51] Fiona C. Denison, Scott I. Semple, Sarah J. Stock, Jane Walker, Ian Marshall, and Jane E. Norman. Novel use of proton magnetic resonance spectroscopy (1HMRS) to non-invasively assess placental metabolism. *PLoS ONE*, 7(8), 2012.

[52] Teresa Victoria, Monica Epelman, Beverly G. Coleman, Steve Horii, Edward R. Oliver, Soroosh Mahboubi, Nahla Khalek, Stefanie Kasperski, J. Christopher Edgar, and Diego Jaramillo. Low-dose fetal CT in the prenatal evaluation of skeletal dysplasias and other severe skeletal abnormalities. *American Journal of Roentgenology*, 200(5):989–1000, 2013.

[53] Michael Tchirikov, Georgios Gatopoulos, Miriam Strohner, Alexander Puhl, and Joscha Steetskamp. Two new approaches in intrauterine tracheal occlusion using an ultrathin fetoscope. *Laryngoscope*, 120(2):394–398, 2010.

[54] Marcel Tella, Pankaj Daga, Francois Chadebecq, Stephen Thompson, Dzhoshkun I. Shakir, George Dwyer, Ruwan Wimalasundera, Jan Deprest, Danail Stoyanov, Tom Vercauteren, and Sebastien Ourselin. A combined em and visual tracking probabilistic model for robust mosaicking: application to fetoscopy. In *CVPR*, pages 524–532, 2016.

[55] Wenfeng Xia, Efthymios Maneas, Daniil I. Nikitichev, Charles A. Mosse, Gustavo Sato Dos Santos, Tom Vercauteren, Anna L. David, Jan Deprest, Sébastien Ourselin, Paul C. Beard, and Adrien E. Desjardins. Interventional photoacoustic imaging of the human placenta with ultrasonic tracking for minimally invasive fetal surgeries. In *Lecture Notes in Computer Science (including subseries*

Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), volume 9349, pages 371–378. 2015.

[56] Minghua Xu and Lihong V. Wang. Photoacoustic imaging in biomedicine. *Review of Scientific Instruments*, 77(4):041101, 2006.

[57] Chloe J. Arthuis, Anthony Novell, Florian Raes, Jean Michel Escoffre, Stephanie Lerondel, Alain Le Pape, Ayache Bouakaz, and Franck Perrotin. Real-time monitoring of placental oxygenation during maternal hypoxia and hyperoxygenation using photoacoustic imaging. *PLoS ONE*, 12(1), 2017.

[58] Irene P Stafford, D Edward Neumann, and Heather Jarrell. Abnormal placental structure and vasa previa: confirmation of the relationship. *Journal of ultrasound in medicine*, 23(11):1521–1522, 2004.

[59] Michael Yampolsky, Carolyn M. Salafia, Oleksandr Shlakhter, Danielle Haas, Barbara Eucker, and John Thorp. Modeling the variability of shapes of a human placenta. *Placenta*, 29:790–797, 2008.

[60] Sonia Dahdouh, Nickie Andescavage, Sayali Yewale, Alexa Yarish, Diane Lanham, Dorothy Bulas, Adre J. Plessis, and Catherine Limperopoulos. In vivo placental MRI shape and textural features predict fetal growth restriction and postnatal outcome. *Journal of Magnetic Resonance Imaging*, 47(2):449–458, 2018.

[61] Hongen Liao, Masayoshi Tsuzuki, Takashi Mochizuki, and Etsuko Kobayashi. Minimally invasive therapy & allied technologies fast image mapping of endoscopic image mosaics with three-dimensional ultrasound image for intrauterine fetal surgery. *Minimally Invasive Therapy & Allied Technologies*, 18(6):332–340, 2009.

[62] Liangjing Yang, Junchen Wang, Etsuko Kobayashi, and Hongen Liao. Ultrasound image-guided mapping of endoscopic views on a 3D placenta model: a tracker-less approach. In *MIAR/AE-CAI*, pages 107–116, 2013.

[63] Gordon N. Stevenson, Sally L. Collins, Jane Ding, Lawrence Impey, and J. Alison Noble. 3-D Ultrasound segmentation of the placenta using the random walker algorithm: reliability and agreement. *Ultrasound in Medicine and Biology*, 41(12):3182–3193, 2015.

[64] Gordon N. Stevenson, Kypros H. Nicolaides, Malid Molloholli, Stavros Natsis, Sally L. Collins, and Childrens Health. Automatic 3D ultrasound segmentation of the first trimester placenta using deep leraning. In *ISBI*, pages 279–282, 2017.

[65] Kio Kim, Piotr A. Habas, Vidya Rajagopalan, Julia A. Scott, James M. Corbett-detig, Francois Rousseau, A. James Barkovich, Orit A. Glenn, Colin Studholme, and Senior Member. Bias field inconsistency correction of motion-scattered multislice MRI for improved 3D image reconstruction. *TMI*, 30(9):1704–1712, 2011.

[66] Bernhard Kainz, Christina Malamateniou, Maria Murgasova, Kevin Keraudren, Mary Rutherford, Joseph V Hajnal, and Daniel Rueckert. Motion corrected 3D reconstruction of the fetal thorax from prenatal MRI. In *MICCAI*, pages 284–291, 2014.

[67] Ali Gholipour, Judy A. Estroff, Simon K. Warfield, and Senior Member. Robust super-resolution volume reconstruction from slice acquisitions : application to fetal brain MRI. *TMI*, 29(10):1739–1758, 2010.

[68] Maria Kuklisova-Murgasova, Gerardine Quaghebeur, Mary A. Rutherford, Joseph V. Hajnal, and Julia A. Schnabel. Reconstruction of fetal brain MRI with intensity matching and complete outlier removal. *Medical image analysis*, 16(8):1550–1564, dec 2012.

[69] Yuanyuan Jia, Ali Gholipour, Zhongshi He, and Simon Warfield. A new sparse representation framework for reconstruction of an isotropic high spatial resolution MR volume from orthogonal anisotropic resolution scans. *TMI*, 36(5):1182–1193, 2017.

[70] Michael Ebner, Karen K. Chung, Ferran Prados, M. Jorge Cardoso, Declan T. Chard, Tom Vercauteren, and Sébastien Ourselin. Volumetric reconstruction from printed films: Enabling 30 year longitudinal analysis in MR neuroimaging. *NeuroImage*, 165:238–250, 2018.

[71] Piotr A. Habas, Kio Kim, James M. Corbett-Detig, Francois Rousseau, Orit A. Glenn, A. James Barkovich, and Colin Studholme. A spatiotemporal atlas of MR intensity, tissue probability and shape of the fetal brain with application to segmentation. *NeuroImage*, 53:460–470, 2010.

[72] Tobias Heimann and Hans Peter Meinzer. Statistical shape models for 3D medical image segmentation: a review. *Medical Image Analysis*, 13(4):543–563, 2009.

[73] Feng Zhao and Xianghua Xie. An overview of interactive medical image segmentation. *Annals of the BMVA*, 2013(7):1–22, 2013.

[74] Guotai Wang, Maria A. Zuluaga, Rosalind Pratt, Michael Aertsen, Tom Doel, Maria Klusmann, Anna L. David, Jan Deprest, Tom Vercauteren, and Sébastien Ourselin. Dynamically balanced online random forests for interactive scribble-based segmentation. In *MICCAI*, pages 352–360, 2016.

[75] Guotai Wang, Maria A. Zuluaga, Rosalind Pratt, Michael Aertsen, Anna L. David, Jan Deprest, Tom Vercauteren, and Sebastien Ourselin. Slic-Seg: Slice-by-slice segmentation propagation of the placenta in fetal MRI using one-plane scribbles and online learning. In *MICCAI*, pages 29–37, 2015.

[76] Guotai Wang, Maria A. Zuluaga, Rosalind Pratt, Michael Aertsen, Tom Doel, Maria Klusmann, Anna L. David, Jan Deprest, Tom Vercauteren, and Sébastien Ourselin. Slic-Seg: A minimally interactive segmentation of the placenta from sparse and motion-corrupted fetal MRI in multiple views. *Medical Image Analysis*, 34:137–147, 2016.

[77] Guotai Wang, Maria A. Zuluaga, Wenqi Li, Rosalind Pratt, Premal A. Patel, Michael Aertsen, Tom Doel, Maria Klusmann, Anna L. David, Jan Deprest,

Sébastien Ourselin, and Tom Vercauteren. DeepIGeoS: A deep interactive geodesic framework for medical image segmentation. *TPAMI*, In press, 2018.

[78] Nida M. Zaitoun and Musbah J. Aqel. Survey on image segmentation techniques. In *Procedia Computer Science*, volume 65, pages 797–806, 2015.

[79] Neeraj Sharma and Lalit M. Aggarwal. Automated medical image segmentation techniques. *Journal of medical physics*, 35(1):3–14, 2010.

[80] Milan Sonka, Vaclav Hlavac, and Roger Boyle. *Image processing, analysis, and machine vision*. Cengage Learning, 2014.

[81] Miss Hetal J. Vala and Astha Baxi. A review on Otsu image segmentation algorithm. *International Journal of Advanced Research in Computer Engineering and Technology*, 2(2):387–389, 2013.

[82] Xiaoyi Jiang and Daniel Mojon. Adaptive local thresholding by verification-based multithreshold probing with application to vessel detection in retinal images. *TPAMI*, 25(1):131–137, 2003.

[83] Mohamed Roushdy. Comparative study of edge detection algorithms applying on the grayscale noisy image using morphological filter. *GVIP journal*, 6(4):17–23, 2006.

[84] Zhao Yu-Qian, Gui Wei-Hua, Chen Zhen-Cheng, Tang Jing-Tian, and Li Ling-Yun. Medical images edge detection based on mathematical morphology. In *EMBC*, volume 6, pages 6492–6495, 2005.

[85] Ari Paasio and Adam Dawidziuk. CNN-based spatio-temporal nonlinear filtering and endocardial boundary detection in echocardiography. *International Journal of Circuit Theory and Applications*, 27(1):171–207, 1999.

[86] Alireza Norouzi, Mohd Shafry Mohd Rahim, Ayman Altameem, Tanzila Saba, Abdolvahab Ehsani Rad, Amjad Rehman, and Mueen Uddin. Medical image segmentation methods, algorithms, and applications. *IETE Technical Review*, 31(3):199–213, 2014.

[87] Alain Tremeau and Nathalie Borel. A region growing and merging algorithm to color segmentation. *Pattern Recognition*, 30(7):1191–1203, 1997.

[88] Frank Y. Shih and Shouxian Cheng. Automatic seeded region growing for color image segmentation. *Image and Vision Computing*, 23(10):877–886, 2005.

[89] Regina Pohle and Klaus D. Toennies. Segmentation of medical images using adaptive region growing. In *SPIE Medical Imaging*, pages 1337–1346, 2001.

[90] Chenyang Xu and Jerry L Prince. Snakes, shapes, and gradient vector flow. *TIP*, 7(3):359–369, 1998.

[91] Michael Kass, Andrew Witkin, and Demetri Terzopoulos. Snakes: active contour models. *IJCV*, 1(4):321–331, 1988.

[92] Stanley Osher and James A. Sethian. Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics*, 79(1):12–49, 1988.

[93] Tenn Francis Chen. Medical image segmentation using level sets. In *PDE and Level Sets: Algorithmic Approaches to Static and Motion Imagery*, pages 301–340. 2008.

[94] Chunming Li, Chenyang Xu, Changfeng Gui, and Martin D. Fox. Distance regularized level set evolution and its application to image segmentation. *TMI*, 19(12):3243–3254, 2010.

[95] Daniel Cremers, Mikael Rousson, and Rachid Deriche. A review of statistical approaches to level set segmentation: integrating color, texture, motion and shape. *IJCV*, 72(2):195–215, 2007.

[96] Timothy F Cootes, Andrew Hill, Christopher J Taylor, and Jane Haslam. Use of active shape models for locating structures in medical images. *Image and Vision Computing*, 12(6):355–366, 1994.

[97] Poay Hoon Lim, Ulas Bagci, and Li Bai. A new prior shape model for level set segmentation. In *Iberoamerican Congress on Pattern Recognition*, pages 125–132, 2011.

[98] Guotai Wang, Shaoting Zhang, Hongzhi Xie, Dimitris N. Metaxas, and Lixu Gu. A homotopy-based sparse representation for fast and accurate shape prior modeling in liver surgical planning. *Medical Image Analysis*, 19(1):176–186, 2015.

[99] Timothy F. Cootes, Gareth J. Edwards, and Christopher J. Taylor. Active appearance models. *TPAMI*, 23(6):681–685, 2001.

[100] Lisa M. Koch, Martin Rajchl, Wenjia Bai, Christian F. Baumgartner, Tong Tong, Jonathan Passerat-Palmbach, Paul Aljabar, and Daniel Rueckert. Multi-atlas segmentation using partially annotated data: methods and annotation strategies. *TPAMI*, 40(7):1683 – 1696, 2017.

[101] Juan Eugenio Iglesias and Mert R. Sabuncu. Multi-atlas segmentation of biomedical images: A survey. *Medical Image Analysis*, 24(1):205–219, 2015.

[102] Hongzhi Wang, Jung W. Suh, Sandhitsu R. Das, John B. Pluta, Caryne Craige, and Paul A. Yushkevich. Multi-atlas segmentation with joint label fusion. In *IPMI*, pages 611–623, 2013.

[103] Da Ma, Manuel J. Cardoso, Marc Modat, Nick Powell, Jack Wells, Holly Holmes, Frances Wiseman, Victor Tybulewicz, Elizabeth Fisher, Mark F. Lythgoe, and Sébastien Ourselin. Automatic structural parcellation of mouse brain MRI using multi-atlas label fusion. *PLoS ONE*, 9(1):e86576, 2014.

[104] Tse-wei Chen, Yi-ling Chen, and Shao-yi Chien. Fast image segmentation based on K-Means clustering with histograms in HSV color space. In *2008 IEEE 10th Workshop on Multimedia Signal Processing*, pages 322–325, 2008.

[105] Zexuan Ji, Jinyao Liu, Guo Cao, Quansen Sun, and Qiang Chen. Robust spa-

tially constrained fuzzy c-means algorithm for brain MR image segmentation. *Pattern Recognition*, 47(7):2454–2466, 2014.

[106] Simon K. Warfield, Kelly H. Zou, and William M. Wells. Validation of image segmentation and expert quality with an expectation-maximization algorithm. In *MICCAI*, pages 298–306, 2002.

[107] Mengyuan Liu, Averi Kitsch, Steven Miller, Vann Chau, Kenneth Poskitt, Francois Rousseau, Dennis Shaw, and Colin Studholme. Patch-based augmentation of Expectation-Maximization for brain MRI tissue segmentation at arbitrary age after premature birth. *NeuroImage*, 127:387–408, 2016.

[108] Jose Dolz, Nacim Betrouni, Mathilde Quidet, Dris Kharroubi, Henri A. Leroy, Nicolas Reyns, Laurent Massoptier, and Maximilien Vermandel. Stacking denoising auto-encoders in a deep network to segment the brainstem on MRI in brain cancer patients: a clinical study. *Computerized Medical Imaging and Graphics*, 52:8–18, 2016.

[109] Guangjun Zhao, Xuchu Wang, Yanmin Niu, Liwen Tan, and Shao Xiang Zhang. Segmenting brain tissues from Chinese visible human dataset by deep-learned features with stacked autoencoder. *BioMed Research International*, 2016:5284586, 2016.

[110] Yanrong Guo, Guorong Wu, Leah A. Commander, Stephanie Szary, Valerie Jewells, Weili Lin, and Dinggang Shen. Segmenting hippocampus from infant brains by sparse patch matching with deep-learned features. In *MICCAI*, pages 308–315, 2014.

[111] Elisa Ricci and Renzo Perfetti. Retinal blood vessel segmentation using line operators and support vector classification. *TMI*, 26(10):1357–1365, 2007.

[112] Leo Breiman. Random forests. *European Journal of Mathematics*, 45:5–32, 2001.

[113] Antonio Criminisi and Jamie Shotton. *Decision forests for computer vision and medical image analysis*. 2013.

[114] Albert Montillo, Jamie Shotton, John Winn, Juan Eugenio Iglesias, Dimitri Metaxas, and Antonio Criminisi. Entangled decision forests and their application for semantic segmentation of CT images. In *IPMI*, pages 184–196, 2011.

[115] Peter Kontschieder, Pushmeet Kohli, Jamie Shotton, and Antonio Criminisi. GeoF geodesic forests for learning coupled predictors. In *CVPR*, pages 65–72, 2013.

[116] Geert Litjens, Thijs Kooi, Babak Ehteshami Bejnordi, Arnaud Arindra Adiyoso Setio, Francesco Ciompi, Mohsen Ghafoorian, Jeroen A.W.M. van der Laak, Bram Van Ginneken, and Clara I. Sánchez. A survey on deep learning in medical image analysis. *Medical Image Analysis*, 42:60–88, 2017.

[117] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105, 2012.

[118] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *CVPR*, pages 1–9, 2015.

[119] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

[120] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[121] Mohammad Havaei, Axel Davy, David Warde-Farley, Antoine Biard, Aaron Courville, Yoshua Bengio, Chris Pal, Pierre-Marc Jodoin, and Hugo Larochelle. Brain tumor segmentation with deep neural networks. *Medical Image Analysis*, 35:18–31, 2016.

[122] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, pages 580–587, 2014.

[123] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, pages 3431–3440, 2015.

[124] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In *MICCAI*, pages 234–241, 2015.

[125] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *TPAMI*, 40(4):834 – 848, 2017.

[126] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. In *ICLR*, 2016.

[127] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected CRFs with gaussian edge potentials. In *NIPS*, pages 109–117, 2011.

[128] Konstantinos Kamnitsas, Christian Ledig, Virginia F. J. Newcombe, Joanna P. Simpson, Andrew D. Kane, David K. Menon, Daniel Rueckert, and Ben Glocker. Efficient multi-scale 3D CNN with fully connected CRF for accurate brain lesion segmentation. *Medical Image Analysis*, 36:61–78, 2017.

[129] Ahmed Abdulkadir, Soeren S. Lienkamp, Thomas Brox, and Olaf Ronneberger. 3D U-Net: Learning dense volumetric segmentation from sparse annotation. In *MICCAI*, pages 424–432, 2016.

[130] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-Net: Fully convolutional neural networks for volumetric medical image segmentation. In *IC3DV*, pages 565–571, 2016.

[131] Wenqi Li, Guotai Wang, Lucas Fidon, Sebastien Ourselin, M. Jorge Cardoso, and Tom Vercauteren. On the compactness, efficiency, and representation of 3D convolutional networks: brain parcellation as a pretext task. In *IPMI*, pages 348–360, 2017.

[132] Rolf Adams and Leanne Bischof. Seeded region growing. *TPAMI*, 16(6):641–647, 1994.

[133] Yuri Y. Boykov and Marie Pierre Jolly. Interactive graph cuts for optimal boundary & region segmentation of objects in N-D images. In *ICCV*, pages 105–112, 2001.

[134] Xue Bai and Guillermo Sapiro. Geodesic matting: a framework for fast interactive image and video segmentation and matting. *IJCV*, 82(2):113–132, 2009.

[135] Leo Grady, Thomas Schiwietz, Shmuel Aharon, and Rüdiger Westermann. Random walks for interactive organ segmentation in two and three dimensions: implementation and validation. In *MICCAI*, pages 773–780, 2005.

[136] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. GrabCut: Interactive foreground extraction using iterated graph cuts. *ACM Trans. on Graphics*, 23(3):309–314, 2004.

[137] Christopher J. Armstrong, Brian L. Price, and William A. Barrett. Interactive segmentation of image volumes with live surface. *Computers and Graphics*, 31(2):212–229, 2007.

[138] Paul A. Yushkevich, Joseph Piven, Heather Cody Hazlett, Rachel Gimpel Smith, Sean Ho, James C. Gee, and Guido Gerig. User-guided 3D active contour segmentation of anatomical structures: significantly improved efficiency and reliability. *NeuroImage*, 31(3):1116–1128, 2006.

[139] William A. Barrett and Eric N. Mortensen. Interactive live-wire boundary extraction. *Medical Image Analysis*, 1(4):331–341, 1997.

[140] Antonio Criminisi, Toby Sharp, and Andrew Blake. GeoS: Geodesic image segmentation. In *ECCV*, pages 99–112, 2008.

[141] Vladimir Vezhnevets and Vadim Konouchine. GrowCut: Interactive multi-label ND image segmentation by cellular automata. In *Graphicon*, pages 150–156, 2005.

[142] Daniel Freedman and Tao Zhang. Interactive graph cut based segmentation with shape priors. In *CVPR*, pages 755–762, 2005.

[143] Wenzhe Shi, Xiahai Zhuang, Robin Wolz, Duckett Simon, Kaipin Tung, Haiyan Wang, Sebastien Ourselin, Philip Edwards, Reza Razavi, and Daniel Rueckert. A multi-image graph cut approach for cardiac image segmentation and uncertainty estimation. In *Statistical Atlases and Computational Models of the Heart. Imaging and Modelling Challenges*, volume 7085, pages 178–187. Springer Berlin Heidelberg, 2012.

[144] Bo Wang, Wei Liu, Marcel Prastawa, Andrei Irimia, Paul M. Vespa, John D. Van Horn, P. Thomas Fletcher, and Guido Gerig. 4D active cut: an interactive tool for pathological anatomy modeling. In *ISBI*, pages 529–532, 2014.

[145] Dhruv Batra, Adarsh Kowdle, Devi Parikh, Jiebo Luo, and Tsuhan Chen. iCoseg: Interactive co-segmentation with intelligent scribble guidance. In *CVPR*, pages 3169–3176, 2010.

[146] Andrew Top, Ghassan Hamarneh, and Rafeef Abugharbieh. Active learning for interactive 3D image segmentation. In *MICCAI*, pages 603–10, 2011.

[147] Harini Veeraraghavan and James V. Miller. Active learning guided interactions for consistent image segmentation with reduced user interactions. In *ISBI*, pages 1645–1648, 2011.

[148] Jakob Santner, Markus Unger, Thomas Pock, Christian Leistner, Amir Saffari, and Horst Bischof. Interactive texture segmentation using random forests and total variation. In *BMVC*, pages 1–12, 2009.

[149] Ignacio Arganda-Carreras, Verena Kaynig, Curtis Rueden, Kevin W. Eliceiri, Johannes Schindelin, Albert Cardona, and H. Sebastian Seung. Trainable weka segmentation: a machine learning tool for microscopy pixel classification. *Bioinformatics*, 33(15):2424–2426, 2017.

[150] Christoph Sommer, Christoph Straehle, Ullrich Kothe, and Fred A. Hamprecht. Ilastik: Interactive learning and segmentation toolkit. In *ISBI*, pages 230–233, 2011.

[151] Imanol Luengo, Michele C. Darrow, Matthew C. Spink, Ying Sun, Wei Dai, Cynthia Y. He, Wah Chiu, Tony Pridmore, Alun W. Ashton, Elizabeth M.H. Duke, Mark Basham, and Andrew P. French. SuRVoS: Super-region volume segmentation workbench. *Journal of Structural Biology*, 198(1):43–53, 2017.

[152] Jingjing Deng, Xianghua Xie, Rob Alcock, and Carl Roobottom. 3D interactive coronary artery segmentation using random forests and Markov random field optimization. In *ICIP*, pages 942–946, oct 2014.

[153] Olga Barinova, Roman Shapovalov, Sergey Sudakov, and Alexander Velizhev. Online random forest for interactive image segmentation. In *EEML*, 2012.

[154] Di Lin, Jifeng Dai, Jiaya Jia, Kaiming He, and Jian Sun. ScribbleSup: Scribble-supervised convolutional networks for semantic segmentation. In *CVPR*, pages 3159–3167, 2016.

[155] Martin Rajchl, Matthew Lee, Ozan Oktay, Konstantinos Kamnitsas, Jonathan Passerat-Palmbach, Wenjia Bai, Mary Rutherford, Joseph Hajnal, Bernhard Kainz, and Daniel Rueckert. DeepCut: Object segmentation from bounding box annotations using convolutional neural networks. *TMI*, 36(2):674–683, 2017.

[156] Ning Xu, Brian Price, Scott Cohen, Jimei Yang, and Thomas Huang. Deep interactive object selection. In *CVPR*, pages 373–381, 2016.

[157] John Lafferty and Andrew Mccallum. Conditional random fields: probabilistic

models for segmenting and labeling sequence data. In *ICML*, volume 2001, pages 282–289, 2001.

[158] Yuri Boykov and Vladimir Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *TPAMI*, 26(9):1124–1137, 2004.

[159] Jing Yuan, Egil Bae, and Xue Cheng Tai. A study on continuous max-flow and min-cut approaches. In *CVPR*, pages 2217–2224, 2010.

[160] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected CRFs. In *ICLR*, 2015.

[161] Vladimir Kolmogorov and Ramin Zabih. What energy functions can be minimized via graph cuts? *TPAMI*, 26(2):147–159, 2004.

[162] Martin Szummer, Pushmeet Kohli, and Derek Hoiem. Learning CRFs using graph cuts. In *ECCV*, pages 582–595, 2008.

[163] Lester Randolph Ford and Delbert Ray Fulkerson. *Flows in networks*, volume 275. 1962.

[164] Andrew V. Goldberg and Robert E. Tarjan. A new approach to the maximum-flow problem. *Journal of the ACM*, 35(4):921–940, 1988.

[165] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *TPAMI*, 23(11):1222–1239, 2001.

[166] Nadia Payet and Sinisa Todorovic. (RF)^2 – random forest random field. In *NIPS*, pages 1885–1893, 2010.

[167] Andrew Adams, Jongmin Baek, and Myers Abraham Davis. Fast high-dimensional filtering using the permutohedral lattice. *Computer Graphics Forum*, 29(2):753–762, 2010.

[168] Jose Ignacio Orlando and Matthew Blaschko. Learning fully-connected CRFs for blood bessel segmentation in retinal images. In *MICCAI*, pages 634–641, 2014.

[169] Justin Domke. Learning graphical model parameters with approximate marginal inference. *TPAMI*, 35(10):2454–2467, 2013.

[170] Philipp Krähenbühl and Vladlen Koltun. Parameter learning and convergent inference for dense random fields. In *ICML*, pages 513–521, 2013.

[171] Shuai Zheng, Sadeep Jayasumana, Bernardino Romera-Paredes, Vibhav Vineet, Zhizhong Su, Dalong Du, Chang Huang, and Philip H. S. Torr. Conditional random fields as recurrent neural networks. In *ICCV*, pages 1529–1537, 2015.

[172] Raviteja Vemulapalli, Oncel Tuzel, Ming-yu Liu, and Rama Chellappa. Gaussian conditional random field network for semantic segmentation. In *CVPR*, pages 3224–3233, 2016.

[173] Fayao Liu, Chunhua Shen, and Guosheng Lin. Deep convolutional neural fields for depth estimation from a single image. In *CVPR*, pages 5162–5170, 2014.

[174] Guosheng Lin, Chunhua Shen, Ian Reid, and Anton van dan Hengel. Efficient piecewise training of deep structured models for semantic segmentation. In *CVPR*, pages 3194–3203, 2016.

[175] Alexander Kirillov, Shuai Zheng, Dmitrij Schlesinger, Walter Forkel, Anatoly Zelenin, Philip Torr, and Carsten Rother. Efficient likelihood learning of a generic CNN-CRF model for semantic segmentation. In *ACCV*, 2016.

[176] Dongfeng Han, John Bayouth, Qi Song, Aakant Taurani, Milan Sonka, John Buatti, and Xiaodong Wu. Globally optimal tumor segmentation in PET-CT images: a graph-based co-segmentation method. In *IPMI*, pages 245–256, 2011.

[177] Fumin Guo, Jing Yuan, Martin Rajchl, Sarah Svenningsen, Dante PI Capaldi, Khadija Sheikh, Aaron Fenster, and Grace Parraga. Globally optimal co-

segmentation of three-dimensional pulmonary 1H and hyperpolarized 3He MRI with spatial consistence prior. *Medical Image Analysis*, 23(1):43–55, 2015.

[178] Anthony Yezzi, Lilla Zöllei, and Tina Kapur. A variational framework for integrating segmentation and registration through active contours. *Medical Image Analysis*, 7:171–185, 2003.

[179] Raphael Prevost, Remi Cuingnet, Benoit Mory, Jean-Michel Correas, Laurent D Cohen, and Roberto Ardon. Joint co-segmentation and registration of 3D ultrasound images. In *IPMI*, pages 268–79, 2013.

[180] Adeline Paiement, Majid Mirmehdi, Xianghua Xie, and Mark C. K. Hamilton. Registration and modeling from spaced and misaligned image volumes. *TIP*, 25(9):4379–4393, 2016.

[181] Dinggang Shen, Guorong Wu, and Heung-Il Suk. Deep learning in medical image analysis. *Annual Review of Biomedical Engineering*, 19(1):1–26, 2017.

[182] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: surpassing human-level performance on ImageNet classification. In *ICCV*, pages 1026–1034, 2015.

[183] John Duchi, Elad Hazan, and Yoram Singer. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. *Journal of Machine Learning Research*, 12:2121–2159, 2011.

[184] Tijmen Tieleman and Geoffrey Hinton. Lecture 6.5-RMSProp, COURSERA: Neural networks for machine learning. Technical report, University of Toronto, 2012.

[185] Diederik P. Kingma and Jimmy Lei Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[186] Sergey Ioffe and Christian Szegedy. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *ICML*, pages 448–456, 2015.

[187] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014.

[188] Jeremie Anquez, Elsa D. Angelini, and Isabelle Bloch. Automatic segmentation of head structures on fetal MRI. In *ISBI*, pages 109–112, 2009.

[189] Ali Gholipour, Alireza Akhondi-Asl, Judy A. Estroff, and Simon K. Warfield. Multi-atlas multi-shape segmentation of fetal brain MRI for volumetric and morphometric analysis of ventriculomegaly. *NeuroImage*, 60:1819–1831, 2012.

[190] Kevin Keraudren, Maria Kuklisova-Murgasova, Vanessa Kyriakopoulou, Christina Malamateniou, Mary A. Rutherford, Bernhard Kainz, Joseph V. Hajnal, and Daniel Rueckert. Automated fetal brain segmentation from 2D MRI slices for motion correction. *NeuroImage*, 101:633–643, 2014.

[191] Seyed Sadegh Mohseni Salehi, Deniz Erdogmus, and Ali Gholipour. Auto-context convolutional neural network (Auto-Net) for brain extraction in magnetic resonance imaging. *TMI*, 36(11):2319 – 2330, 2017.

[192] Kevin Keraudren, Bernhard Kainz, Ozan Oktay, Vanessa Kyriakopoulou, Mary Rutherford, Joseph V. Hajnal, and Daniel Rueckert. Automated localization of fetal organs in MRI using random forests with steerable features. In *MICCAI*, pages 620–627, 2015.

[193] Tong Zhang, Jacqueline Matthew, Maelene Lohezic, Alice Davidson, Mary Rutherford, Daniel Rueckert, Joseph Hajnal, and Paul Aljabar. Graph-based whole body segmentation in fetal MR images. In *MICCAI workshop on PIPPI*, 2016.

[194] Maria Kuklisova-Murgasova, Amalia Cifor, Raffaele Napolitano, Aris Papageorghiou, Gerardine Quaghebeur, Mary A. Rutherford, Joseph V. Hajnal, J. Alison Noble, and Julia A. Schnabel. Registration of 3D fetal neurosonography and MRI. *Medical Image Analysis*, 17(8):1137–1150, 2013.

[195] Amir Alansary, Konstantinos Kamnitsas, Alice Davidson, Martin Rajchl, Christina Malamateniou, Mary Rutherford, Joseph V. Hajnal, Ben Glocker, Daniel Rueckert, and Bernhard Kainz. Fast fully automatic segmentation of the human placenta from motion corrupted MRI. In *MICCAI*, pages 589–597, 2016.

[196] Taghi M. Khoshgoftaar, Moiz Golawala, and Jason Van Hulse. An empirical study of learning from imbalanced data using random forest. In *ICTAI*, pages 310–317, 2007.

[197] Chao Chen, Andy Liaw, and Leo Breiman. *Using random forest to learn imbalanced data*. PhD thesis, University of California, Berkeley, 2004.

[198] João V. B. Soares, Jorge J. G. Leandro, Roberto M. Cesar Júnior, Herbert F. Jelinek, and Michael J. Cree. Retinal vessel segmentation using the 2-D Gabor wavelet and supervised classification. *IEEE transactions on medical imaging*, 25(9):1214–1222, 2006.

[199] Xiangrong Zhang, Licheng Jiao, Fang Liu, Liefeng Bo, and Maoguo Gong. Spectral clustering ensemble applied to SAR image segmentation. *IEEE Transactions on Geoscience and Remote Sensing*, 46(7):2126–2136, 2008.

[200] David A. Clausi. An analysis of co-occurrence texture statistics as a function of grey level quantization. *Canadian Journal of Remote Sensing*, 28(1):45–62, 2014.

[201] Amara Graps. Introduction to wavelets. *IEEE computational science & engineering*, 2(2):50–61, 1995.

[202] Kamarul Hawari Ghazali, Mohd Fais Mansor, Mohd. Marzuki Mustafa, and Aini Hussain. Feature extraction technique using discrete wavelet transform for image classification. In *2007 5th Student Conference on Research and Development*, pages 1–4, 2007.

[203] Paul Viola and Michael J. Jones. Robust real-time face detection. *IJCV*, 57(2):137–154, 2004.

[204] Wei-Yin Loh. Classification and regression trees. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 1(1):14–23, 2011.

[205] Laura Elena Raileanu and Kilian Stoffel. Theoretical comparison between the Gini Index and Information Gain criteria. *Annals of Mathematics and Artificial Intelligence*, 41(1):77–93, 2004.

[206] Amir Saffari, Christian Leistner, Jakob Santner, Martin Godec, and Horst Bischof. Online random forests. In *ICCV*, pages 1393–1400, 2009.

[207] Hien M. Nguyen, Eric W. Cooper, and Katsuari Kamei. Online learning from imbalanced data streams. In *SoCPaR*, pages 347–352, 2011.

[208] Adel Ghazikhani, Reza Monsefi, and Hadi Sadoghi Yazdi. Ensemble of online neural networks for non-stationary and imbalanced data streams. *Neurocomputing*, 122:535–544, 2013.

[209] Miroslav Kubat and Stan Matwin. Addressing the Curse of Imbalanced Training Sets: One Sided Selection. *ICML*, 97:179–186, 1997.

[210] Guotai Wang, Maria A Zuluaga, Rosalind Pratt, Michael Aertsen, Anna L. David, Jan Deprest, Tom Vercauteren, and Sébastien Ourselin. Minimally interactive placenta segmentation from motion corrupted MRI for fetal surgical planning. In *MICCAI Workshop on Interactive Medical Image Computing*, 2015.

[211] Terry S. Yoo. The insight toolkit: An open-source initiative in data segmentation and registration. In *Visualization Handbook*, pages 733–748. 2005.

[212] Daniel Rueckert, Luke I. Sonoda, Carmel Hayes, Derek LG Hill, Martin O. Leach, and David J. Hawkes. Nonrigid registration using free-form deformations: application to breast MR images. *TMI*, 18(8):712–21, 1999.

[213] Marc Modat, Gerard R. Ridgway, Zeike A. Taylor, Manja Lehmann, Josephine Barnes, David J. Hawkes, Nick C. Fox, and Sébastien Ourselin. Fast free-form deformation using graphics processing units. *Computer Methods and Programs in Biomedicine*, 98(3):278–284, 2010.

[214] Joseph L. Fleiss. Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76(5):378–382, 1971.

[215] Ron Kikinis and Steve Pieper. 3D Slicer as a tool for interactive brain tumor segmentation. In *EMBS*, pages 6982–6984, 2011.

[216] Wenjia Bai, Wenzhe Shi, Declan P. O'Regan, Tong Tong, Haiyan Wang, Shahnaz Jamil-Copley, Nicholas S. Peters, and Daniel Rueckert. A probabilistic patch-based label fusion model for multi-atlas segmentation with registration refinement: application to cardiac MR images. *TMI*, 32(7):1302–1315, 2013.

[217] Pierre Sermanet, David Eigen, Xiang Zhang, Michael Mathieu, Rob Fergus, and Yann LeCun. OverFeat: Integrated recognition, localization and detection using convolutional networks. In *ICLR*, 2014.

[218] Holger R. Roth, Le Lu, Amal Farag, Hoo-chang Shin, Jiamin Liu, Evrim B. Turkbey, and Ronald M. Summers. DeepOrgan: Multi-level deep convolutional networks for automated pancreas segmentation. In *MICCAI*, pages 556–564, 2015.

[219] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. In *ACMICM*, pages 675–678, 2014.

[220] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009.

[221] Xue Bai and Guillermo Sapiro. A geodesic framework for fast interactive image and video segmentation and matting. In *ICCV*, pages 1–8, 2007.

[222] Leo Grady. Random walks for image segmentation. *TPAMI*, 28(11):1768–1783, 2006.

[223] Guotai Wang, Wenqi Li, Maria A. Zuluaga, Rosalind Pratt, Premal A. Patel, Michael Aertsen, Tom Doel, Anna L. David, Jan Deprest, Sebastien Ourselin, and Tom Vercauteren. Interactive medical image segmentation using deep learning with image-specific fine-tuning. *TMI*, PP(99):1–1, 2018.

[224] Hebert Luchetti Ribeiro and Adilson Gonzaga. Hand image segmentation in video sequence by GMM: a comprarative analysis. In *SIBGRAPI*, pages 357–364, 2006.

[225] Guotai Wang, Wenqi Li, Sébastien Ourselin, and Tom Vercauteren. Automatic brain tumor segmentation using cascaded anisotropic convolutional neural networks. In *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries*, pages 178–190. Springer International Publishing, 2018.

[226] Georgios Zoumpourlis, Alexandros Doumanoglou, Nicholas Vretos, and Petros Daras. Non-linear Convolution Filters for CNN-Based Learning. In *ICCV*, volume 2017-Octob, pages 4771–4779, 2017.

[227] Eli Gibson, Wenqi Li, Carole Sudre, Lucas Fidon, Dzhoshkun I. Shakir, Guotai Wang, Zach Eaton-Rosen, Robert Gray, Tom Doel, Yipeng Hu, Tom Whyntie, Parashkev Nachev, Marc Modat, Dean C. Barratt, Sébastien Ourselin, M. Jorge Cardoso, and Tom Vercauteren. NiftyNet: A deep-learning platform for medical imaging. *Computer Methods and Programs in Biomedicine*, 158:113–122, 2018.

[228] Esmitt Ram and Pablo Temoche. A volume segmentation approach based on GrabCut. *CLEI Electronic Journal*, 16(2):4–4, 2013.

[229] Bjoern H. Menze, Andras Jakab, Stefan Bauer, Jayashree Kalpathy-Cramer, Keyvan Farahani, Justin Kirby, Yuliya Burren, Nicole Porz, Johannes Slotboom, Roland Wiest, Levente Lanczi, Elizabeth Gerstner, Marc André Weber, Tal Arbel, Brian B. Avants, Nicholas Ayache, Patricia Buendia, D. Louis

Collins, Nicolas Cordier, Jason J. Corso, Antonio Criminisi, Tilak Das, Hervé Delingette, Çaatay Demiralp, Christopher R. Durst, Michel Dojat, Senan Doyle, Joana Festa, Florence Forbes, Ezequiel Geremia, Ben Glocker, Polina Golland, Xiaotao Guo, Andac Hamamci, Khan M. Iftekharuddin, Raj Jena, Nigel M. John, Ender Konukoglu, Danial Lashkari, José António Mariz, Raphael Meier, Sérgio Pereira, Doina Precup, Stephen J. Price, Tammy Riklin Raviv, Syed M.S. Reza, Michael Ryan, Duygu Sarikaya, Lawrence Schwartz, Hoo Chang Shin, Jamie Shotton, Carlos A. Silva, Nuno Sousa, Nagesh K. Subbanna, Gabor Szekely, Thomas J. Taylor, Owen M. Thomas, Nicholas J. Tustison, Gozde Unal, Flor Vasseur, Max Wintermark, Dong Hye Ye, Liang Zhao, Binsheng Zhao, Darko Zikic, Marcel Prastawa, Mauricio Reyes, and Koen Van Leemput. The multimodal brain tumor image segmentation benchmark (BRATS). *TMI*, 34(10):1993–2024, 2015.

[230] Lucas Fidon, Wenqi Li, Luis C. Garcia-Peraza-Herrera, Jinendra Ekanayake, Neil Kitchen, Sebastien Ourselin, and Tom Vercauteren. Scalable multimodal convolutional networks for brain tumour segmentation. In *MICCAI*, pages 285–293, 2017.

[231] Nima Tajbakhsh, Jae Y. Shin, Suryakanth R. Gurudu, R. Todd Hurst, Christopher B. Kendall, Michael B. Gotway, and Jianming Liang. Convolutional neural networks for medical image analysis: full training or fine tuning? *TMI*, 35(5):1299–1312, 2016.

[232] Yarin Gal and Zoubin Ghahramani. Dropout as a Bayesian approximation: representing model uncertainty in deep learning. In *ICML*, pages 1050–1059, 2016.

[233] Yoshua Bengio, Aaron C. Courville, and Pascal Vincent. Unsupervised feature learning and deep learning: a review and new perspectives. *CoRR*, abs/1206.5, 2012.

[234] Guorong Wu, Minjeong Kim, Qian Wang, Yaozong Gao, Shu Liao, and Ding-

gang Shen. Unsupervised deep feature learning for deformable registration of MR brain images. In *MICCAI*, pages 649–656, 2013.

[235] Hoo-Chang Chang Shin, Matthew R. Orton, David J. Collins, Simon J. Doran, and Martin O. Leach. Stacked autoencoders for unsupervised feature learning and multiple organ detection in a pilot study using 4D patient data. *TPAMI*, 35(8):1930–1943, 2013.

[236] Sébastien Tourbier, Xavier Bresson, Patric Hagmann, Jean Philippe Thiran, Reto Meuli, and Meritxell Bach Cuadra. An efficient total variation algorithm for super-resolution in fetal brain MRI with adaptive regularization. *NeuroImage*, 118:584–597, 2015.

[237] Michael Ebner, Manil Chouhan, Premal A. Patel, David Atkinson, Zahir Amin, Samantha Read, Shonit Punwani, Stuart Taylor, Tom Vercauteren, and Sébastien Ourselin. Point-spread-function-aware slice-to-volume registration: application to upper abdominal MRI super-resolution. In *Reconstruction, Segmentation, and Analysis of Medical Images: First International Workshops, RAMBO 2016 and HVSMR 2016*, pages 3–13. 2017.

[238] Carolyn M Salafia, Michael Yampolsky, Dawn P Misra, Oleksander Shlakhter, Barbara Eucker, and John Thorp. Placental surface shape, function, and effects of maternal and fetal vascular pathology. *Placenta*, 31(11):958–962, 2011.

[239] Kenji Suzuki, Hiroyuki Abe, Heber MacMahon, and Kunio Doi. Image-processing technique for suppressing ribs in chest radiographs by means of massive training artificial neural network (MTANN). *TMI*, 25(4):406–416, 2006.

[240] Bram van Ginneken, Mikkel B. Stegmann, and Marco Loog. Segmentation of anatomical structures in chest radiographs using supervised methods: a comparative study on a public database. *Medical Image Analysis*, 10(1):19–40, 2006.

[241] Laurens Hogeweg, Clara I. Sánchez, Pim A. de Jong, Pragnya Maduskar, and Bram van Ginneken. Clavicle segmentation in chest radiographs. *Medical Image Analysis*, 16(8):1490–1502, 2012.

[242] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, Xiaoqiang Zheng, and Google Brain. TensorFlow: A system for large-scale machine learning. In *OSDI*, pages 265–284, 2016.